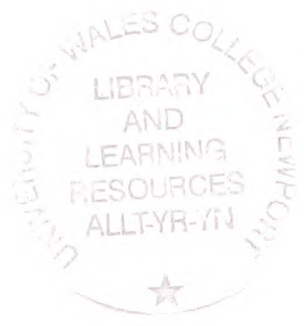

Fuzzy Filters for depth map smoothing

Thesis submitted to the University of Wales for the degree of

Doctor of Philosophy

By

Neil Rothwell Hughes, BSc, MSc
Department of Engineering
University of Wales College, Newport
1999

Declarations

DECLARATION

This work has not previously been accepted in any substance for any degree and is not being currently submitted in candidature for any degree

Signed Neil Bethell Hughes ... (candidate)

Date 6th August 1999

STATEMENT 1

This thesis is the result of my own investigations except where otherwise stated.
Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed Neil Bethell Hughes. (candidate)

Date 6th August 1999

STATEMENT 2

I hereby give consent for my thesis, if accepted to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed Neil Bethell Hughes (candidate)

Date 6th August 1999

Acknowledgements

I would like to thank Professor Geoff Roberts for his friendly advice, support, and encouragement and especially his patience during the period of my studies. Thanks also to Dr. Graham Wilson for his advice and support during the earlier part of the project, and also to my fellow students, especially the pioneers of B14. Especially thanks to Mary for her love and support without whom none of this would seem possible.

Summary

This thesis is concerned with the extraction of dense three-dimensional depth maps from sequences of two-dimensional greyscale images using correlation based matching. In particular the thesis is focused on the noise processes that occur in the depth map and the removal of that noise using nonlinear filters based on fuzzy systems.

The depth from stereo algorithm is reviewed and a widely used correlation based matcher, the Sum Squared Difference (SSD) matcher, is introduced together with an established method of measuring sub-pixel disparities in stereo pairs of images. The noise in the disparity map associated with this matcher is investigated. The conjecture is made that a fuzzy inferencing system can be trained to perform a nonlinear filtering process which is more effective than conventional filters at removing the mixed impulsive and Gaussian-like noise present in the depth map.

Six methods of training fuzzy systems of the Sugeno type based on the simulated annealing algorithm are proposed and tested by training fuzzy systems to approximate a simple function of two variables

The thesis reviews existing fuzzy logic based filters and proposes a taxonomy for such systems. This distinguishes between direct and indirect acting fuzzy filters. An indirect acting fuzzy filter is applied to the task of smoothing a disparity map. The first order Sugeno fuzzy system is then proposed as an architecture that would be suitable as the basis for a direct acting fuzzy filter. This architecture is then applied to the task of smoothing depth maps derived from real and simulated data.

The main contributions of the thesis are the identification of the Sugeno fuzzy system as a form of filter, the proposed training techniques, and the application of fuzzy filters to depth map smoothing.

Table of Contents

List of Figures

List of Tables

Glossary of Terms

Chapter 2

(u,v)	Image space co-ordinates
f	Focal length
C	Optical centre
(x, y, z)	Object or world space co-ordinates
$[U,V,S]$, **M**	Two-dimensional projective space co-ordinates
$[X,Y,Z,S]$, **W**	Three-dimensional projective space co-ordinates
P	Projection matrix.
P'	Projection matrix adjusted for origin shift and image co-ordinate scaling
(u_0,v_o)	Image co-ordinate system origin
k_u and k_v	u and v co-ordinate scaling factors
α_u, α_v	Product of k_u and k_v with f
E	Euclidean rotation and translation matrix
ω, ϕ, κ	Angles of tilt, pan, and swing
(t_x, t_y, t_z)	Position of optical centre in image co-ordinate system
P''	Projection matrix including Euclidean shift of origin and rotation of co-ordinate axes.
θ	Angle between image plane co-ordinate axes.
P'''	Projection matrix allowing for non-orthogonal image plane co-ordinate axes
$\mathbf{T}_s$	Matrix describing camera motion.
P	Pixel side dimension
$I(u ,v, ,t)$	Brightness of image point at time t
$d_{[n]}$	Disparity in pixels at camera step n
$g_{[n]}$	Zero mean Gaussian noise corrupting grey scale image
σ_g^2	Variance of grey scale image noise
$M_{[n]}$	Measured disparity at step n.
C	Kalman filter measurement constant
$v_{[n]}$	Zero mean Gaussian noise due to disparity measurement process
σ_v^2	Variance of measurement process noise
$\hat{d}_{[n]}$	Estimated disparity at step *[n]*
$P_{[n]}$	Mean squared error estimate of Kalman filter at step *[n]*
$K_{[n]}$	Kalman filter gain at step *[n]*

Chapter 3

$L(u, v)$, $R(u, v)$	Grey level values of points in a pair of images
SSD	Sum of squared differences value
$n(u,v)$	Corrupting Gaussian noise at pixel (u,v)
σ_n^2	Variance of corrupting Gaussian noise
$F(u,v)$	True grey level value at pixel (u,v)
$d(u, v)$	True disparity for pixel (u,v)
a,b,c	Coefficients of quadratic fitted to SSD values for different estimates of disparity
S_1	SSD value at position 1
d_0	Nearest integer disparity value to fitted quadratic minimum
Q(d)	Quadratic fitted to three disparity versus SSD points
d_{minpos}	Sub-pixel disparity value at fitted quadratic minimum

P_f	Probability distribution of grey scale values
P(SD)	Probability distribution for the square of the difference of any two pixels
Sum , Diff	Random variables defined by equation 3.24 and 3.25
P_{jsd}	Joint probability of random variables *Diff* and *Sum*
Pj(.)	Joint probability distribution
$P_s(.)$	Marginal probability distribution
J_{gg}	Jacobian defined in equation 3.29
P_{my},	Probability distribution of the sub-pixel disparities
P_D	Distribution of disparities
P_z	Distribution of depths
Δt_x	Camera translation along the x-axis
c(x)	Noise-corrupted output signal of an unconstrained process
s(x)	Possible *s*olution for *c(x)*
E	Energy or error functional to be minimised
$S(\omega)$, $C(\omega)$	Fourier transforms of *s(x), c(x)* .

Chapter 4

$\hat{x}[i]$	Output of FIR filter
y[i]	Input to FIR filter
h[i]	Impulse response of FIR filter
E[.]	Expected value operator
ε	Mean squared error
(i,j), (p,q)	Pixel position indices
$R_{xx}(.)$	Autocorrelation function
ω_k , ω_l	Two-dimensional frequencies
$H[\omega_k , \omega_l]$	Two-dimensional frequency transfer function
$PSD_x[\omega_k , \omega_l]$	Power spectral density of signal
$PSD_n[\omega_k , \omega_l]$	Power spectral density of noise
$med(\mathbf{x}_n)$,	Median of a vector **x** of length n
$\mathbf{x}_{ranked}$	Vector formed by rank ordering the *n* samples in the vector $\mathbf{x}_n$
T(**x**)	Location estimator
F	True distribution of observations
IF(x;T,F)	Influence function
G_{n-1}	Histogram of n-1 observations
V(T,F)	Asymptotic variance
ARE(T,S)	Asymptotic relative efficiency

Chapter 5

$\star$	t-norm operator
$\oplus$	t-conorm operator
$\mu_A(x)$	Fuzzy set membership function
R	fuzzy or crisp relation
Min(.)	Minimum operator
Max(.)	Maximum operator
Sup(.)	Supremum operator
$R_O S$	Composition formed by union of two fuzzy relations R and S
$\mu_{A\rightarrow B}(x,y)$	Membership function defining an implication operator
τ	Height of fuzzy singleton
Gauss(.)	Gaussian membership function

μ_f	Location parameters of Gaussian membership function
σ_f	Width parameters of Gaussian membership function
w_r', w_r	Unormalised and normalised rule firing weights
$\boldsymbol{X}$	Vector of crisp input values
$\boldsymbol{A}_r$	Output set parameter vector for the r^{th} rule
W	Vector of firing weights
T	Training data input matrix
$\mathbf{O}_t$	Training data vector of outputs
$\boldsymbol{P}$	Vector of all output set parameters
$\boldsymbol{O}$	D by 1 output vector for D training data sets
$\boldsymbol{B}^{\Diamond}$	Pseudo inverse of a matrix **B**
E_r	Energy for a system in microstate r
K	Boltzmann constant
Z	Partition function for a system
T	Temperature parameter
UOD_lo	Lower bound of universe of discourse
UOD _hi	Upper bound of universe of discourse
OOBCost(n,f)	Out of bounds error for the n^{th} input and the f^{th} Gaussian input fuzzy set
R_0	Radius parameter for out of bound error cost
Trapz(a,b,c,d)	Trapezoidal membership function with parameters *a, b, c, d*
Metrop	Function defined by equation 5.27
α	Cooling coefficient
T[0]	Initial value of temperature parameter

Chapter 6

k(i,j)	Macroinformation variable
Var(i,j)	Sample variance within a window centred on pixel (i,j)
Pixel dist	Distance of pixel from filter window centre
d_{median}	Median of disparities within a filter window
disparity_difference	Measure of difference of a pixel disparity value from filter window median
O_i	i^{th} output fuzzy set
Dfil(i,j)	Filtered output of indirect fuzzy filter at pixel i,j.
a_i	Sugeno output set coefficients
W_F	Rule firing weights
$\boldsymbol{X}$	Input data vector
$f(\mathbf{X})$	Function performed by Sugeno fuzzy system which maps input vector to firing weights

Chapter 1 : Introduction and Overview of Thesis

1.1 Introduction

This thesis is concerned with an aspect of the important problem of the recovery of three-dimensional information from sequences of two-dimensional images. The recovery of the lost depth information is an example of a class of problems termed 'inverse problems'. Such problems are often mathematically 'ill-posed' such that the solution does not exist, is not unique, or is unstable to small perturbations in the data. Many of the image processing techniques used to recover depth information are ill-posed (Bertero *et al*, 1988) and the result of this is that the depth data that is recovered contains large errors which appear as sharp noise spikes in the recovered depth map. The removal of these noise spikes without also removing important information from the depth map poses a difficult challenge. This challenge can be regarded as a filtering problem or equivalently as a problem of regularising (*op cit.* Bertero *et al*, 1988) the ill-posed depth recovery problem by applying a priori knowledge about the allowable forms of real depth maps.

The main focus and contribution of this thesis is to put forward and test a new approach to filtering dense depth maps using techniques based on the Sugeno (Takagi and Sugeno, 1983) fuzzy inferencing method. The use of fuzzy logic was adopted as an approach to filtering for two reasons. Firstly, a fuzzy system can approximate any mapping from input data to output data, which allows fuzzy systems to offer the possibility of performing any arbitrary linear or non-linear filtering operation. Secondly, a fuzzy system offers a possible means of capturing the a priori regularising constraints in a particular non-linear mapping in an explicit way through its rulebase. The new fuzzy filter structure, which is introduced in the thesis, can also be applied to other problem domains such as filtering mixed noise in one-dimensional signals that contain discontinuities.

The fuzzy filter structure can also be optimised or trained using exemplar data and a further important contribution made in this thesis is to apply the stochastic search technique known as simulated annealing to the optimisation of fuzzy systems of the Sugeno type. Simulated annealing was explored as a training method because unlike the more commonly used gradient descent techniques simulated annealing as a search technique is not prone to becoming trapped in local minima.

The thesis also makes two further small contributions by analysing the noise processes associated with correlation-based matching, which is widely used in depth map generation, and by suggesting an approach to classifying fuzzy logic-based filters.

1.2 Background to the Problem

Since the very earliest days of electronic computers researchers have looked to the possibility of creating entities which display that characteristic of human beings that is called intelligence. The beginnings of the serious scientific quest for 'artificial intelligence' dates therefore from the time that machines capable of implementing Boolean logic reasoning processes on a large scale became available. Since the landmark essay of Turing (Turing, 1950), the debate about whether a large capacity finite state machine can mimic human intelligence has raged. Moreover, the quest for artificial intelligence has been constantly encouraged by speculation that its goal can be achieved with the next generation of increased computational power.

With the search for artificial intelligence came a debate on the exact nature and definition of intelligence. This debate has led to the identification of a collection of attributes, at least some of which are required to be present before an entity's behaviour can be said to be intelligent. These attributes include the ability to sense and analyse the immediate environment, to learn patterns in the environment, and to predict by inference the future behaviour of the environment.

The ability to sense and learn represents a necessary requirement before intelligent behaviour can be said to be present. An intelligent entity must also place itself within, and include itself as part of the environment, whilst at the same time distinguishing itself from the environment. This ability to self locate within an environment is necessary for any mobile entity to navigate that environment. This navigational 'self-awareness' may be the origins of that other attribute of intelligence called consciousness. Consciousness may therefore be an emergent property of the intelligence needed for mobility and navigation. Certainly, life forms that are not mobile, and do not need therefore to navigate, do not display any of the attributes of intelligent behaviour.

A major feature of naturally occurring intelligence is purposive autonomous motion. In nature such behaviour is ultimately directed towards enhanced genetic success. A restricted goal for the creator of artificial intelligence might be limited autonomous behaviour in a machine directed towards achieving a task that would be too dangerous or costly to carry out using human beings or remote control. An example of this might be a roving vehicle designed to explore the surface of a remote planet in the solar system. Another example might be an autonomous robot whose task would be to continuously inspect the underwater structure of an oilrig. Such limited goals are termed 'soft' artificial intelligence to distinguish them from the higher goal of creating an entity displaying 'hard' artificial intelligence which displays the full human attributes of consciousness, creativity, and emotion.

For an intelligence to exhibit itself through autonomous behaviour it needs the ability to extract information about the environment around it. Indeed, the ability to sense its environment may be essential to the development of such intelligence. For this reason the ability to make explicit certain properties of a scene, called low level vision, may form an important adjunct to creating artificially intelligent machines. The identification of the relative (and absolute) depth of points in a scene is one such low level vision process. This low level vision task is also important for

industrial inspection, robot guidance and autonomous vehicle navigation. It is to aid in the task of depth reconstruction that the new class of fuzzy logic based filters described in this thesis have been developed. However, although the fuzzy based filters have been motivated from the requirements of low level depth reconstruction and developed with this task in mind, they are applicable to other filtering problems.

The basic principle of operation of the technique used to generate depth maps that is explored in this research is that of correlation based stereo matching. A simplifying assumption is made that the viewed scene is assumed to be static. A pair of images is taken with each image in the pair being taken from different camera positions, but with identical cameras. The stereo matching approach to depth map generation from image pairs can be broken down into the following steps:

- Points are identified in the two images that represent the same point in the viewed scene. These are called corresponding points.

- The shift in the position of the corresponding points between the two images is measured. This shift is known as the disparity.

- The image formation process for the camera is determined. This consists of obtaining the geometrical transformation or projection from three-dimensional object space to the two-dimensional image space. This step is called the camera calibration process and is described by a camera projection matrix.

- From knowledge of the camera projection matrix and the disparity, the original three-dimensional co-ordinates of the point in the viewed scene can be determined.

- The above steps are repeated for a sampled set of points in the images. In this way a depth map of the viewed scene is generated. This depth map is a two-dimensional function of the image co-ordinates and is relative to a fixed origin defined in the calibration step.

The camera calibration problem is relatively well understood (Tsai, 1987). The most difficult step is that of establishing corresponding points. The simplest method of establishing correspondence is to use a correlation-based matcher. The Sum of Squared Differences (SSD) matcher (Anandan, 1984) is an example of such a correlation matcher and is used in the work described in this thesis. It can be used to produce dense depth maps and can be adapted to give a measure of the uncertainty in the measured depth (Matthies *et al*, 1989). Feature based matching is another powerful matching technique which can give more reliable results than correlation based matching. However the resulting depth maps are sparse and require interpolation between the points for which a depth value is determined.

An examination of depth maps generated from synthesised and real images using the SSD matcher reveals that they are corrupted by noise and areas of gross error. As noted in section 1.1, these gross errors can be attributed to the fact that the matching process is an example of an ill-posed inverse problem. In this case the problem is that of inverting a projection or mapping from three-dimensional space to two-dimensional space so as to recover the lost information of the third dimension. The classic method of solving such ill-posed problems and eliminating the gross errors is called regularisation. In order to regularise a problem a prior condition or conditions are set which limit the range of allowable solutions. The prior conditions are chosen so that they represent a known fact about the nature of the true solution to the problem. Such prior conditions are often described by an energy functional or cost penalty, which penalises solutions that violate the prior assumptions. The measured data is also included in the energy functional as a term that penalises solutions which are different from that suggested by the data. The solution finally adopted is such as to strike a balance between that suggested by the measured data and that which most closely satisfies the prior assumption. The regularised

problem is then cast a minimisation problem for which the solution is that which minimises the energy functional. The minimisation of the functional can be carried out using iterative gradient descent methods or more robust search techniques but both of these approaches can be computationally intensive.

The prior assumptions that are used in regularisation are derived from knowledge of the world that the solution describes. A common prior constraint that is applied to depth maps is that the depth map should be generally smooth. This constraint reflects reality in general but is violated at a few points in a viewed scene where depth discontinuities exist. Unfortunately these depth discontinuities occur at edges. The locations of these edges represent the most important information in a depth map. In other words, the prior constraint fails at the most important points in the depth map. It is the combination of this complexity in the real world, and hence the possible solution space, and the ill-posed nature of the problem of depth recovery from two-dimensional images which lends it its fascination and challenge.

Certain simple energy functionals that are commonly used to regularise the depth recovery problem express depth map smoothness in terms of the depth map gradients. For these cases the minimisation of the energy functional can be carried out analytically using the calculus of variations (Terzopoulos, 1986(b)). The analysis shows that the minimisation is equivalent to passing the data through a linear filter. However, the effect of a simple linear filter on depth map edges is to severely round them and introduce uncertainties as to their location. This also suggests that an energy functional that allows for depth discontinuities is likely to be equivalent to a non-linear filter of some form.

As well as being viewed as the consequence of the ill-posed nature of the problem, the observed gross errors can be viewed as a form of unwanted noise corrupting the wanted signal of the depth map. An analysis of this noise reveals that the noise has the characteristics of mixed

Gaussian and impulsive noise. The normal operation of the SSD matcher results in a Gaussian noise process. However, where the matcher makes a false match or fails to make a match the result is a spike or impulse in the depth map. The task of regularising the ill-posed problem can be viewed as a filtering problem in which the filter is required to remove impulses and smooth Gaussian noise, whilst at the same time preserving edges in the depth map.

Order statistic filters form a class of non-linear filters which are effective in the task of removing impulsive noise, whilst preserving edges. The median filter is the most widely used of this class of filter. However, these filters are suboptimal for Gaussian noise. A broad task, which is an active area of current research, is to create non-linear filters that are effective in filtering mixed noise containing a strong impulsive component, but which preserve discontinuities in the data.

Fuzzy logic was first described as a technique for decision making in the presence of uncertainty by Zadeh (Zadeh 1965), (Zadeh, 1973). Since then its use in control engineering has grown because of its effectiveness in controlling complex and non-linear systems. A particular advantage is the use that the designer of a system using fuzzy logic can make of 'expert' knowledge expressed in linguistic terms. A fuzzy system maps input values to output values in a way which is determined by a set of linguistically expressed rules. This mapping can be complex and non-linear. One view of fuzzy systems is to regard them as Black box universal function approximators (Wang, 1992), (Kosko, 1994). The Black box can be trained using exemplar data to closely match the desired non-linear mapping. Since the non-linear mapping does not have to be explicitly described, but is instead captured through the exemplar data, such universal function approximator systems are sometimes termed 'model free estimators'.

The main contribution to knowledge of this thesis is to develop a class of non-linear filter based on a fuzzy logic paradigm. The motivation for the development of this new fuzzy filter was the need to smooth dense depth maps without using computationally slow iterative techniques. Thus the 'fuzzy filters' have been designed and trained to tackle the problem of filtering depth maps generated using correlation-based matching and which, therefore, are corrupted by mixed Gaussian and impulsive noise. The body of reported work on the use of soft computing approaches to non-linear filtering contains far more papers on filters based on neural networks than papers using a fuzzy logic based approach. Nevertheless, fuzzy inferencing systems share the universal function approximator property of artificial neural networks (*op cit.* Kosko, 1994), and moreover posses an advantage in their ability to capture expert knowledge. Of the relatively few papers which report the use of fuzzy logic based filters, the vast majority use the fuzzy logic stage as a 'bolt on' means of controlling a conventional linear or non-linear filter. A rarely adopted approach (Kim and Kosko, 1996) is to perform all the filtering within the fuzzy logic stage. These two approaches are designated respectively 'indirect' and 'direct' in this thesis. The new fuzzy filter structure proposed and explored in this thesis is based on the Takagi-Sugeno fuzzy inferencing system. This fuzzy filter structure is a direct filter structure but can also be related to indirect fuzzy filter structures. It is believed that the work described in this thesis is the first time that a Sugeno inferencing system has been used as the basis for a filtering structure, as distinct from as a structure for a control system. It is also believed that the thesis contains the first explicit classification of fuzzy filters as direct or indirect acting. Furthermore it is believed that no previously reported work on fuzzy filters specifically address the problem of depth map filtering so that the work reported in this thesis represents a new application for fuzzy logic based filtering.

The new fuzzy filter structure can also be trained using exemplar input output data, which is a model-free approach to filter design. In the context of filtering viewed from a regularisation standpoint, a fuzzy or neural network filtering system generated using the model-free approach

captures the prior assumptions implicit in the data. This makes the choice of representative data critical. Investigation of the general properties of such filters may require that data be modelled in order to create synthetic data whose properties are well understood. A potential disadvantage of the model-free approach is that the action of the filter is not explicitly described and therefore not necessarily understood. This may not be a problem if the exemplar data has itself been explicitly modelled, in which case the action of the trained filter is understood, albeit at one remove. The fact that fuzzy systems contain rules that capture linguistic statements about the desired action of the filter allows fuzzy systems to be more explicit and therefore, potentially, to be better understood than neural network based filters. Unlike neural network based filters, it is possible to 'take the lid off the Black Box' in a trained fuzzy system. However, this statement about the relative explicitness of fuzzy systems must not be made too glibly. Much depends on the structure of the fuzzy system and the details of the algorithm used to train it and also on how that fuzzy system is integrated into an overall scheme for filtering.

In order to take full advantage of the ability of the fuzzy systems to form model-free estimators, efficient and effective training techniques are needed. Fuzzy systems can be created using expert knowledge alone to create the rulebase. The input set parameters can be set initially so as to evenly cover the input universe of discourse. The input set parameters can then be manually or automatically adjusted on exemplar data to minimise the error between a desired output and the fuzzy system's actual output. This adjustment has to be of a limited nature if the original meaning of the rulebase is to be maintained. Unlimited variation of input set parameters results in a change in the meaning of a rule. For example a rulebase may contain the predicates "If X is small then ....." and "If X is large then....". The meaning of these two predicates changes if the input sets are changed so that "small" becomes larger than "large".

The price to be paid for unlimited training of input fuzzy sets is, therefore, a loss of the explicit understanding of the fuzzy system's action. However, unlimited training may result in a fuzzy

system that performs better than a system trained using constrained training. This is especially true for non-exhaustive rulebases that do not contain all the possible rules for a given set of inputs and input fuzzy sets. It is frequently necessary to use non-exhaustive rulebases when fuzzy systems are used as the basis for two–dimensional filters. This is because the problem of dimensionality (*op cit* Kim and Kosko, 1996) comes into play. This problem is the exponential rise in the size of the exhaustive rulebase with the number of inputs. What is not explicitly mentioned by them is that the hypervolume to be explored by a training algorithm that adjusts the input set parameters also increases with input vector dimensionality and that it also does so exponentially. Exponential increase in the computational size of a problem with scaling leads to a computational load which is termed Non-polynomial (NP) complete and is the hallmark of a computationally intractable problem. Non-exhaustive rulebases are one answer to this problem and the use of non-exhaustive rulebases is discussed in Chapter Five. With the use of non-exhaustive rulebases a new training option arises. This is the option of selecting which subset of all the possible rules should be used in the rulebase. Taken all together there are three possible things to adjust whilst training a Sugeno inferencing system. These are the input set parameters, the output set parameters and the choice of rules in the sub rulebase.

The input and output parameters of a Sugeno fuzzy system can be optimised for a given set of input data by using a combination of linear least squares and backpropagation (Jang, 1993), the output parameters being optimised by the linear least squares and the input parameters by the backpropagation. However the backpropagation algorithm is a form of gradient descent, and can become trapped in local minima of the n-dimensional error versus input parameter surface. The simulated annealing algorithm (Metropolis *et al.*, 1953), (Kirkpatrick *et al.*, 1983) is designed to optimise combinatorial problems whilst avoiding such local minima, and it can be modified for use in the optimisation of problems with continuous variables (Vanderbuilt and Louie, 1984), (Press *et al.*, 1994). This thesis describes the application of simulated annealing in combination with linear least squares for the optimisation of input and output parameters in

Sugeno fuzzy systems in general, and of Sugeno-based fuzzy filters in particular. It is believed that this work is the first reported use of simulated annealing for training Sugeno inferencing systems, although there is a recent report of the use simulated annealing to optimise a Mamdani type fuzzy model (Garibaldi and Ifeachor, 1999). The thesis also describes the use of simulated annealing for the combinatorial problem of choosing an optimal non-exhaustive rule base.

It is interesting that the iterative techniques used for filtering or regularising depth maps by previous workers in the field have used search techniques that are related to simulated annealing as the basis for their regularisation algorithm (Geman and Geman, 1984), (Acton, 1997). In trying to create filters based on fuzzy logic as a one-pass alternative to these iterative approaches, the need for a search technique to train the fuzzy filters has been identified. In this thesis the same search technique, namely simulated annealing has been used. Other search techniques, notably genetic algorithm approaches might also be tried in future. The important difference between the iterative approach and the one pass non-linear filtering based on fuzzy logic approach is that the searching is done off-line in the fuzzy approach.

1.3 Overview of the thesis

The structure of the remainder of the thesis follows broadly that of the description of the problem in section 1.2 above, starting with a review of the stereo matching process and the noise associated with it and leading on to a review of possible methods of filtering the depth maps. The thesis goes on to review fuzzy techniques and describes the new approaches proposed for the training of Sugeno fuzzy systems using simulated annealing, before introducing the use of fuzzy systems as filters. The new Sugeno-based fuzzy filter is then introduced and the results of applying it to one and two-dimensional real and simulated depth maps are presented. Finally the conclusions from the work described in the thesis are drawn and recommendations for future work are made.

Chapter Two describes in more detail the steps necessary for stereo matching and the camera calibration problem is discussed.

Chapter Three continues with an analysis and tests on the performance of the sum of squared difference matching technique. This analysis shows that the noise in the depth map consists of mixed Gaussian and impulsive noise. The Gaussian component of the noise is shown to be a consequence of the interaction of the sub-pixel SSD matcher and the statistics of the underlying intensity patterns of the image, whilst the impulsive component of the noise is shown to occur when the intensity patterns in the image cause the matcher to make a false match.

Chapter Four contains a brief review of existing major approaches to linear and non-linear filter structures, including the Wiener and median filter which are used as benchmarks with which to compare the performance of the novel filters described in this thesis.

Chapter Five of this thesis contains an overview of different approaches to fuzzy inferencing. It also contains a discussion of the suitability of different inferencing systems as approximators for different types of mappings or functions. Chapter Five also describes the new approaches to fuzzy system training using the simulated annealing algorithm. Within this chapter six different ways of training a Sugeno fuzzy inferencing system are described and the results of testing these training methods on a simple function approximation problem are presented. Amongst the training methods is a method for automatically selecting an optimal non-exhaustive rulebase. The fuzzy system training software, which was written to test the ideas on fuzzy training algorithms put forward in this thesis, is also described in this chapter.

Chapter Six discusses various existing approaches to the use of fuzzy inferencing in filtering problems. It includes a detailed review of the literature on fuzzy logic in filtering and also

introduces a taxonomy of fuzzy filters based on the way that the fuzzy system is integrated into the overall filtering scheme. This comparison of reported work on fuzzy logic-based filters provides a context within which two possible filter structures for depth map smoothing are introduced. The first structure is based on an indirect filter structure and a zeroth order Sugeno fuzzy system. The second structure is the new filter structure based on a first order Sugeno network and this structure is described in detail.

Chapter Seven presents the results of applying the fuzzy filters described in Chapter Six and trained using the methods of Chapter Five to one-dimensional signals and simulated two-dimensional depth maps corrupted by mixed noise. The results of applying the filters to depth maps generated from simulated image data, and to real depth maps generated from real image sequence data are also presented here. Comparisons are also made between the effect of Wiener and median filters and the fuzzy filters when tested on simulated data. The results of some comparisons between the effects of non-exhaustive and exhaustive rulebases are also made in Chapter Seven.

Finally Chapter Eight reviews the thesis, summarises the conclusions from the work presented in the thesis and makes recommendations for further work. Appendix A contains copies of three conference papers that have been produced during the course of the work described in the thesis.

1.4 References

(Acton, 1997) Acton S. T. "Image Restoration Using Generalised Deterministic Annealing." Digital Signal Processing Vol 7 part 2 pp 94-104. 1997

(Anandan, 1984) Anandan P., "Computing dense displacement fields with confidence measures in scenes containing occlusion. " Proceedings DARPA image understanding workshop. pp 236-246, 1984.

(Bertero *et al*, 1988) Bertero M, Poggio T and Torre V, "Ill Posed Problems in Early Vision", Proceedings IEEE , 76 , pp869 – 889 1988 .

(Garibaldi and Ifeachor, 1999) Garibaldi J. M. and Ifeachor E.C. "Application of Simulated Annealing Fuzzy Model Tuning to Umbilical Cord Acid-Base Interpretation." IEEE Transactions on Fuzzy Systems Vol 7. No 1 pp 72- 84. Feb 1999.

(Geman and Geman, 1984) Geman S. and Geman D. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 6 No. 6.

(Jang, 1993) Jang J-S. R. "ANFIS Adaptive-Network-Based Fuzzy Inference System." IEEE Transactions on Systems Man and Cybernetics, Vol 23 No. 3. May 1993.

(Kim and Kosko, 1996) Kim H., and Kosko B. "Fuzzy prediction and filtering in impulsive noise." Fuzzy Sets and Systems Vol. 77 No.1. pp 15-33. January 1996.

(Kirkpatrick *et al.*, 1953) Kirkpatrick S., Gelatt C. D., and Vecchi M. "Optimisation by Simulated Annealing." Science. Vol 220, Issue 4598. pp 671-680. May 1983.

(Kosko, 1994) Kosko B "Fuzzy systems as universal approximators." IEEE Transactions on Computing Vol. 43 pp 1329-1333. 1994.

(Matthies *et al.*, 1989) Matthies L., Kanade T.,and Szeliski R., "Kalman Filter-based Algorithms for Estimating Depth from Image Sequences." International Journal of Computer Vision. No3, pp 209-236, 1989.

(Metropolis *et al.*1953) Metropolis N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H., Teller E. "Equation of State Calculations by Fast Computing Machines" Journal of Chemical Physics Vol 21, No 6. pp 1087-1092. June 1953.

(Press *et al.*, 1994) Press W.H., Teukolsky S.A., Vetterling W.T., and Flannery B.P. "Numerical Recipes in C The Art of Scientific Computing"2nd Edition C.U.P. 1994.

(Takagi and Sugeno, 1983) Takagi T and Sugeno M, "Derivation of Fuzzy Control Rules from Human Operator's Control Actions." Proceedings IFAC Symposium on Fuzzy Information Knowledge Representation and Decision Analysis , pp 55-60. 1983.

(Terzopoulos 1986(b)) Terzopoulos D. "Regularisation of Inverse Visual Problems Involving Discontinuities." IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 8 No. 4 pp 413 - 424, July 1986.

(Tsai, 1987) Tsai R. "A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses." IEEE Journal of Robotics and Automation. Vol RA-3 No. 4 August 1987.

(Turing, 1950) Turing A. "Can a Machine Think?" Mind. Vol. 59. O.U.P. 1950.

(Vanderbuilt and Louie, 1984) Vanderbuilt D. and Louie S. G. "A Monte Carlo Simulated Annealing approach to Optimisation over Continuous Variables." Journal of Computational Physics Vol. 56 pp 259-271, 1984.

(Wang, 1992) Wang L-X. "Fuzzy Systems are Universal Approximators." Proceedings IEEE International Conference on Fuzzy Systems San Diego pp 1163-1169. 1992

(Zadeh 1965). Zadeh L. "Fuzzy Sets." Information and Control Vol. 8 pp 338-353, 1965

(Zadeh, 1973) Zadeh L., "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes", IEEE Transactions on Systems Man and Cybernetics Vol. 3 No. 1 , 28-44 , January 1973.

Chapter 2: Recovery of Depth from Image Sequences

2.1 Introduction

The purpose of this chapter is to outline the theory behind the stereo matching technique used in this thesis to extract depth from image sequences. The chapter begins with a review of the formation of grey-scale images by a camera via the perspective projection and goes on to discuss how depth information can be recovered from pairs of such grey-scale images. As stated in Chapter 1, recovery of the depth information by stereopsis requires the identification of corresponding points in pairs of images and the measurement of the relative displacement or disparity of these points between the two images. This chapter, therefore, goes on to review two broad approaches to measuring the disparity, the correlation-based matchers and the optic flow approach. A correlation-based technique due to (Anandan, 1984), called the Sum of Squared differences (SSD), allows a sub-pixel estimation of the disparity and a measure of the uncertainty associated with the disparity. This estimate and the uncertainty measure can be input to a Kalman filter (Kalman, 1960) to allow the fusion of depth estimates from a sequence of multiple pairs of images (Matthies *et al.*, 1989). The fusion of the depth information from multiple images by the Kalman filter should result in a temporal (or sequence number) filtering of the noise in the depth map provided that the noise is essentially Gaussian and that the uncertainty estimate produced by the matcher is accurate and consistent. Chapter 3 examines in detail the theory and noise performance of the matchers proposed by (*op cit* Anandan, 1984)and (*op cit* Matthies *et al.*, 1989).

2.2 Perspective and inverse perspective projections

2.2.1 3-D to 2-D projection

An imaging system such as a camera or the human eye forms a two-dimensional image of the three-dimensional world. For monochrome camera systems, a point in the three-dimensional object space maps to an intensity or brightness value in the two dimensional image space. The brightness value at a point in the two-dimensional space is a function of the incident light intensity and angle as well as the reflectance of the object at the corresponding point in three-dimensional space. The two dimensional co-ordinates of a point in the image space which corresponds to a point in the object space depends on the three-dimensional co-ordinates of the object space point through a mapping called a projection. In order to recover the three-dimensional co-ordinates of a point in object space from the measured two-dimensional image space co-ordinates, it is necessary to derive a model that captures the projection from object to image space.

The model that is used to model the three to two-dimensional projection is called the 'pinhole camera' model. The geometric behaviour of many image-forming systems, including complex systems with multi-element lenses, can be quite accurately modelled by the pinhole camera model (Trucco and Verri, 1998). The geometric optical approximation (rectilinear propagation of light) is used by the model, which assumes that all the light rays from an object in the three dimensional world space which impinge on the planar sensing surface or retina pass through a point C called the optical centre. The optical centre can be thought of as a small hole cut in an opaque screen, hence the pinhole camera model name. The optical centre lies in a plane called the focal plane of the camera, and the retina lies in the image plane. The distance from focal to image plane is f which is

the focal length of the optical system. The geometry of the pinhole camera model is shown in Figure 2.1.

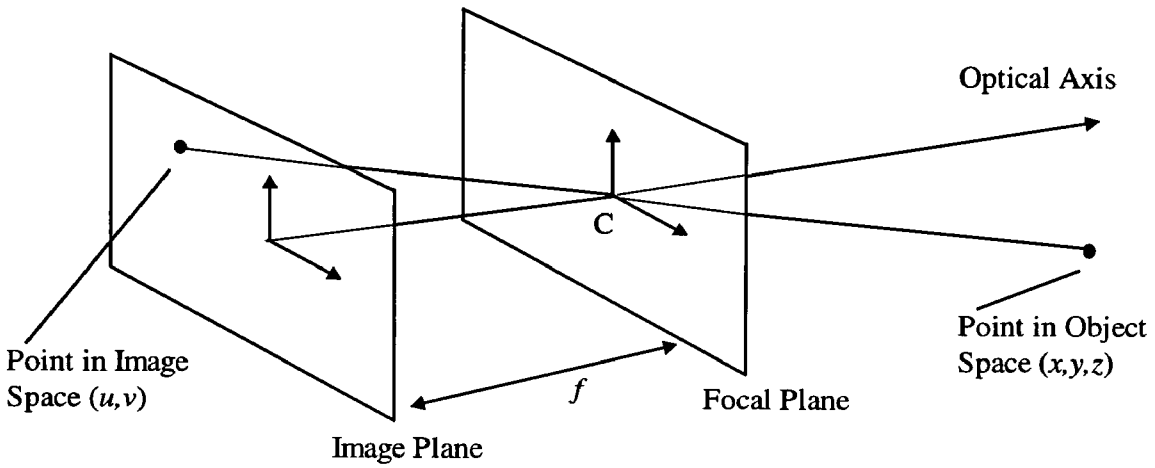

Figure 2.1 Geometry of the Pinhole camera model

The position of a point in the image space can be described by two Cartesian co-ordinates (u, v) relative to an image space origin and that of a point in object space by three co-ordinates (x, y, z) relative to an object space origin. In order to describe the projection from object to image space, it is necessary to obtain a relationship between the co-ordinates (x, y, z) of the object space point that maps to an image space point of co-ordinates (u, v).

It can be assumed without loss of generality that the origin of the object space Cartesian co-ordinate system is at the pinhole C, and the origin of the two-dimensional image space co-ordinate system is at the point of intersection of the perpendicular to the focal plane at C with the image plane. The length of this perpendicular is designated f. By similar triangles, the image space co-ordinates and object space co-ordinates are related by:

$$\frac{u}{x} = \frac{v}{y} = -\frac{f}{z} \qquad 2.1$$

This relationship is often expressed in the homogenous co-ordinate form (Haralick and Shapiro, 1993):

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad 2.2$$

With:

$$U = S.u \qquad 2.3$$

and

$$V = S.v \qquad 2.4$$

Equation 2.2 expresses the image co-ordinates in terms of the co-ordinates $[U, V, S]^T$ of a two dimensional projective space, P^2. The negative sign of the 'f' terms in the equations show the inversion of the image that occurs under pinhole camera projection. If the object space co-ordinates are expressed in terms of a three dimensional projective space, P^3, then equation 2.2 can be written:

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ T \end{bmatrix} \qquad 2.5$$

Or in the matrix form:

$$\mathbf{M} = \mathbf{P}.\mathbf{W} \qquad 2.6$$

Equations 2.5 and 2.6 express a camera model that performs a projective transformation from the three-dimensional object projective space to the two-dimensional image projective plane. The advantage of expressing the camera model in this way is that equation 2.6 is a linear relation whereas the more intuitive expression of the model in equation 2.1 involves non-linear equations. The matrix **P** is the projection matrix. Changes to the image plane origin and scaling and to the world space origin can be accommodated by suitably modifying **P**. Given the co-ordinates (x, y, z) of a point in the three dimensional world, the resulting image space co-ordinates, (u, v), can be computed from equation 2.5 and equations 2.3 and 2.4.

Straight lines in the three-dimensional world plot to lines in the image plane (*op cit.* Haralick and Shapiro, 1993). The vector equation of a straight line passing through points $\mathbf{r_1} = (x_1, y_1, z_1)^T$ and $\mathbf{r_2} = (x_2, y_2, z_2)^T$ in the three-dimensional world is:

$$\mathbf{r} = \lambda\mathbf{r_1} + (1-\lambda)\mathbf{r_2} \qquad 2.7$$

where λ is some scalar for every point on the line. By applying equations 2.2 to 2.6 the equation for the straight line in image space corresponding to the line described by equation 2.7 is given by:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \eta\begin{pmatrix} u \\ v_1 \end{pmatrix} + (1-\eta)\begin{pmatrix} u_2 \\ v_2 \end{pmatrix} \qquad 2.8$$

Where:

$$\eta = \frac{\lambda z_1}{\lambda z_1 + (1-\lambda) z_2} \qquad 2.9$$

2.2.2 2-D to 3-D inverse projection

Equation 2.5 with equations 2.3 and 2.4 allow the computation of the image co-ordinates of a given three dimensional point. The inverse problem is that of determining the three dimensional co-ordinates *(x, y, z)* of a given image point *(u, v)*. The ray projecting *(x, y, z)* to *(u, v)* must pass through the point C = (0, 0, 0) and the point *(u, v, -f)*. The vector equation of a line passing through two given points is given by equation 2.7 so that in Cartesian co-ordinates for this case:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \lambda \begin{pmatrix} u \\ v \\ -f \end{pmatrix} + (1-\lambda) \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \lambda \begin{pmatrix} u \\ v \\ -f \end{pmatrix} \qquad 2.10$$

Therefore any point on the line defined by equation 2.10 projects to *(u, v)* and so, in contrast to the forward projection, the inverse projection of a point is a line. The perspective projection from three-dimensional object space to two-dimensional image space results in a loss of information. For a point, the information that is lost is the z co-ordinate or depth of the point.

2.2.3 Image plane to camera plane transformation

The projection matrix **P**, which has been described in section 2.2.1, has a world origin at the optical centre C, and an image plane origin at the centre of the image plane and such that the line joining the image and world origins is perpendicular to both the focal and image plane. It has also been

assumed that the u and v co-ordinate axis scaling is the same. In a real imaging system, it is convenient to have an image plane origin either at the bottom left of the imaged area (after allowing for inversion) for Cartesian co-ordinates or at the top left of the viewed image when dealing with the image in terms of matrix element addressing. In either case a shift of image co-ordinate system origin is required. A further complication with real cameras is that the individual sensing elements of the camera are not square so that the u and v axis scaling after sampling is different. Both of these complications can be dealt with by adjusting the projection matrix **P.** If the origin of the old co-ordinate system is at (u_0, v_o) in the new co-ordinate system then the co-ordinates of a pixel in the new co-ordinate system (u_{new}, v_{new}) are given in terms of the old co-ordinates, (u_{old}, v_{old}) by:

$$u_{new} = u_{old} + u_0 \tag{2.11}$$

And

$$v_{new} = v_{old} + v_0 \tag{2.12}$$

The difference in scaling between the u and v co-ordinates can be allowed for by introducing two scaling factors k_u and k_v (Faugeras, 1993) so that equations 2.11 and 2.12 become:

$$u_{new} = k_u u_{old} + u_0 \tag{2.13}$$

And:

$$v_{new} = k_v v_{old} + v_0 \tag{2.14}$$

These changes can be incorporated into the projection matrix **P,** which becomes:

$$\mathbf{P}' = \begin{bmatrix} -k_u f & 0 & u_0 & 0 \\ 0 & -k_v f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad 2.15$$

The terms $k_u f$ and $k_v f$ can be written as α_u and α_v. The parameters α_u, α_v, u_0, and v_0 are known as the intrinsic parameters of the camera as they depend only on the camera itself rather than its position or orientation. The parameter f is related to the focal length of the imaging system, but varies as objects at different distances are brought into focus.

The effect of moving the object space co-ordinate system origin from the optical centre and the orientation of the object space axes in space can be allowed for by a Euclidean rotation and translation. Such a rotation and translation is defined by a matrix (*op cit.* Faugeras, 1993) where:

$$\mathbf{E} = \begin{pmatrix} r_{11} r_{12} r_{13} t_x \\ r_{21} r_{22} r_{23} t_y \\ r_{31} r_{32} r_{33} t_z \\ 0 \; 0 \; 0 \; 1 \end{pmatrix} \quad 2.16$$

And:

$$r_{11} = \cos\phi \cos\kappa \quad 2.17$$

$$r_{12} = \sin\omega \sin\phi \cos\kappa + \cos\omega \sin\kappa \quad 2.18$$

$$r_{13} = -\cos\omega \sin\phi \cos\kappa + \sin\omega \sin\kappa \quad 2.19$$

$$r_{21} = -\cos\phi \sin\kappa \quad 2.20$$

$$r_{22} = -\sin\omega \sin\phi \sin\kappa + \cos\omega \cos\kappa \quad 2.21$$

$$r_{23} = \cos\omega \sin\phi \sin\kappa + \sin\omega \cos\kappa \quad 2.22$$

$$r_{31} = \sin\phi \quad 2.23$$

$$r_{32} = -\sin\omega\cos\phi \qquad 2.24$$

$$r_{33} = \cos\omega\cos\phi \qquad 2.25$$

Where the angle of rotation about the x-axis (tilt) is denoted ω, the angle about the y-axis (pan) ϕ, and the angle about the z-axis (swing) κ. The position of the old camera co-ordinate system origin (the optical centre, C) in the new co-ordinate system is (t_x, t_y, t_z). The new overall projection matrix $\mathbf{P}''$ is obtained by post-multiplying the projection matrix $\mathbf{P}'$ by the matrix $\mathbf{E}$ to give:

$$\mathbf{P}'' = \begin{pmatrix} \alpha_u r_{11} + u_0 r_{31} & \alpha_u r_{12} + u_0 r_{32} & \alpha_u r_{13} + u_0 r_{33} & \alpha_u t_x + u_0 t_z \\ \alpha_v r_{21} + v_0 r_{31} & \alpha_v r_{22} + v_0 r_{32} & \alpha_v r_{23} + v_0 r_{33} & \alpha_v t_y + v_0 t_z \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix} \qquad 2.26$$

The twelve parameters r_{ii}, t_x, t_y, and t_z are called the extrinsic parameters of the camera and depend on the position of the camera relative to the external co-ordinate system being used. There are, however, constraints on the matrices which can be written in the form of 2.26 (Horn, 1986), (*op cit.* Faugeras, 1993). These constraints arise from the orthonormality of the 3 by 3 submatrix defined by r_{ii} as given by equations 2.17 to 2.25 and have the result that the determination of the projection matrix $\mathbf{P}''$ from measurements of image and object co-ordinates becomes a non-linear problem.

2.2.4 Camera calibration

Camera calibration consists of determining the projection matrix $\mathbf{P}''$ and from this determining the intrinsic and extrinsic parameters. Equation 2.26 has six unknown extrinsic parameters since the nine rotation parameters r_{ii} are determined using equations 2.17 to 2.25 from the three angles ω, ϕ, and κ. Together with the four unknown intrinsic parameters, there are therefore at least ten parameters to be determined in order to calibrate the camera. For extraction of depth information

using stereo techniques, however, it is not necessary to determine all the extrinsic and intrinsic parameters. The parameters that need to be determined for stereopsis are discussed in section 2.3.

Estimation of the projection matrix begins with the measurement of N reference points in three dimensional space, $\mathbf{M}_i = (x_I, y_i, z_i)$, together with their corresponding co-ordinates in the image plane $\mathbf{m}_i = (u_i, v_i)$ with (i=1...N). From equation 2.6 using $\mathbf{P}''$:

$$\mathbf{M} = \mathbf{P}''.\mathbf{W} \qquad 2.27$$

If the elements of the matrix $\mathbf{P}''$ are written in the form q_{ii} in order to simplify the appearance of the matrix, then $\mathbf{P}''$ becomes:

$$\mathbf{P}'' = \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \end{pmatrix} \qquad 2.28$$

The relationship between the image co-ordinates (u_i, v_i) and the object co-ordinates, (x_i, y_i, z_i) becomes:

$$S.u_i = q_{11}x_i + q_{12}y_i + q_{13}z_i + q_{14} \qquad 2.29$$

$$S.v_i = q_{21}x_i + q_{22}y_i + q_{23}z_i + q_{24} \qquad 2.30$$

$$S_i = q_{31}x_i + q_{32}y_i + q_{33}z_i + q_{34} \qquad 2.31$$

So that:

$$q_{11}x_i + q_{12}y_i + q_{13}z_i + q_{14} - q_{31}u_ix_i - q_{32}u_iy_i - q_{33}u_iz_i - q_{34}u_i = 0 \qquad 2.32$$

$$q_{21}x_i + q_{22}y_i + q_{23}z_i + q_{24} - q_{31}v_ix_i - q_{32}v_iy_i - q_{33}v_iz_i - q_{34}v_i = 0 \qquad 2.33$$

Equations 2.32 and 2.33 can be written as a linear homogenous matrix equation:

$$\mathbf{A}.\mathbf{q} = 0 \qquad 2.34$$

Where **A** is a 2N by 12 matrix dependent on x_i, y_i, z_i, u_i, and v_i and **q** is the 12 x1 column vector of parameters q_{11} ....q_{34} . If **A** is of rank 12 then only the meaningless solution **q** = **0** exists. In general it can be shown (*op cit.* Faugeras, 1993) that if the number of calibration points N ≥ 6 then rank **A**=11, and a unique solution exists. However, as mentioned in section 2.2.3 above, not all values of q_{ii} fit the camera model of equation 2.26. Specifically the model of equation 2.26 requires that:

$$q_{31}^{\ 2} + q_{32}^{\ 2} + q_{33}^{\ 2} = 1 \qquad 2.35$$

And:

$$(q_{11}, q_{12}, q_{13}) \wedge (q_{31}, q_{32}, q_{33}) \bullet (q_{21}, q_{22}, q_{23}) \wedge (q_{31}, q_{32}, q_{33}) = 0 \qquad 2.36$$

(where $\wedge$ and $\bullet$ indicate the vector and scalar products of the vectors (q_{i1}, q_{i2}, q_{i3}))

The second condition as expressed by equation 2.36 is not necessary if the projection matrix $\mathbf{P}''$ is modified so as to allow the axes of the image plane to be non-orthogonal. This introduces a parameter θ, which is the angle between the u and v axes in the image plane. The projection matrix becomes $\mathbf{P}'''$ where:

$$\mathbf{P}''' =$$

$$\begin{pmatrix} \alpha_u r_{11} + u_0 r_{31} \ldots . & \alpha_u r_{12} + u_0 r_{32} \ldots & \alpha_u r_{13} + u_0 r_{33} \ldots & \alpha_u t_x + u_0 t_z \\ \ldots . - \dfrac{\alpha_u r_{21}}{\tan\theta} & \ldots . - \dfrac{\alpha_u r_{22}}{\tan\theta} & \ldots . - \dfrac{\alpha_u r_{23}}{\tan\theta} & \ldots . - \dfrac{\alpha_u t_z}{\tan\theta} \\ \dfrac{\alpha_v r_{21}}{\sin\theta} + v_0 r_{31} & \dfrac{\alpha_v r_{22}}{\sin\theta} + v_0 r_{32} & \dfrac{\alpha_v r_{23}}{\sin\theta} + v_0 r_{33} & \dfrac{\alpha_v t_y}{\sin\theta} + v_0 t_z \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix} \quad 2.37$$

With this camera model the parameters q_{ii} can be found by minimising:

$$\min_{q} \left\| \mathbf{A}.\mathbf{q} \right\|^2 \quad 2.38$$

Subject to the constraint of equation 2.35.

2.3 Depth recovery from multiple pairs of images

2.3.1 Description of technique

As outlined in section 2.2.2, knowledge of the depth of a point on an object in a viewed scene is lost under perspective projection onto a two dimensional image. The depth information can be recovered, however, by taking two or more views of the same scene from different camera viewpoints whose relative positions and orientations are known. If the displacement of corresponding image points between such a pair of images is determined and using the known

projection matrix then the depth co-ordinate and hence the three dimensional co-ordinates of the scene point can be recovered.

If a point $\mathbf{W}= (X,Y,Z,T)^T$ expressed in homogenous co-ordinates is imaged by a camera from two known positions 1 and 2, using equation 2.27, assuming that the camera rows and columns are orthogonal, and if the motion from position 1 to position 2 is described by the Euclidean transformation matrix $\mathbf{T}_s$ the corresponding image plane co-ordinates for positions 1 and 2, $\mathbf{M_1}$ and $\mathbf{M_2}$ are given by:

$$\mathbf{M}_1 = \mathbf{P}''.\mathbf{W} \tag{2.39}$$

$$\mathbf{M}_2 = \mathbf{P}''\mathbf{T}_s.\mathbf{W} \tag{2.40}$$

So that:

$$\mathbf{M}_1 - \mathbf{M}_2 = \mathbf{P}''\mathbf{W} - \mathbf{P}''\mathbf{T}_s.\mathbf{W} = \mathbf{P}''(\mathbf{I} - \mathbf{T}_s)\mathbf{W} \tag{2.41}$$

In general the transformation matrix $\mathbf{T}_s$ has the same form as the matrix $\mathbf{E}$ in equation 2.16. However, if it is assumed that the camera motion from position 1 to position 2 involves no further rotation relative to the axes but is simply a translation, then $\mathbf{T}_s$ is given by:

$$\mathbf{T}_s = \begin{pmatrix} 1 & 0 & 0 & \Delta t_x \\ 0 & 1 & 0 & \Delta t_y \\ 0 & 0 & 1 & \Delta t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.42}$$

Then:

$$\mathbf{I}-\mathbf{T}_s = \begin{pmatrix} 0 & 0 & 0 & -\Delta t_x \\ 0 & 0 & 0 & -\Delta t_y \\ 0 & 0 & 0 & -\Delta t_z \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad 2.43$$

$$\begin{pmatrix} U_1 \\ V_1 \\ S_1 \end{pmatrix} - \begin{pmatrix} U_2 \\ V_2 \\ S_2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & -\Delta t_x(\alpha_u r_{11} + u_0 r_{31}) - \Delta t_y(\alpha_u r_{12} + u_0 r_{32}) - \Delta t_z(\alpha_u r_{13} + u_0 r_{33}) \\ 0 & 0 & 0 & -\Delta t_x(\alpha_v r_{21} + v_0 r_{31}) - \Delta t_y(\alpha_v r_{22} + v_0 r_{32}) - \Delta t_z(\alpha_v r_{23} + v_0 r_{33}) \\ 0 & 0 & 0 & -\Delta t_x r_{31} - \Delta t_y r_{32} - \Delta t_z r_{33} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix} \qquad 2.44$$

By careful mounting of the camera and the CCD array it can be arranged that the angles κ, ω, and φ are zero or close to zero so that 2.44 becomes:

$$\begin{pmatrix} U_1 \\ V_1 \\ S_1 \end{pmatrix} - \begin{pmatrix} U_2 \\ V_2 \\ S_2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & -\Delta t_x \alpha_u - \Delta t_z u_0 \\ 0 & 0 & 0 & -\Delta t_y \alpha_v - \Delta t_z v_0 \\ 0 & 0 & 0 & -\Delta t_z \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix} \qquad 2.45$$

If it is further assumed that the camera is translated in the x-y plane only then $\Delta t_z = 0$, and:

$$\begin{pmatrix} \Delta U \\ \Delta V \\ \Delta S \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & \alpha_u \Delta t_x \\ 0 & 0 & 0 & \alpha_v \Delta t_y \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix} \qquad 2.46$$

Where: $\Delta U = U_2 - U_1$ etc. As $T = 1$ and $S = z$, the change in the image co-ordinates is given by:

$$\Delta u = -\frac{1}{z} \alpha_u \Delta t_x \qquad 2.47$$

$$\Delta v = -\frac{1}{z}\alpha_v \Delta t_y \qquad 2.48$$

Under the above assumptions camera motion along the x-axis alone results in image plane motion along the u-axis. Similarly, motion along the y-axis results in image plane motion along the v-axis. By determining the shift of corresponding points in the image plane, under known camera motion, the depth, or z co-ordinate of those corresponding points can be determined using equations 2.47 and 2.48. The shift in the image plane of corresponding points is often known as the 'disparity'. A two-dimensional disparity map for all points in the image plane can be built-up and converted to an equivalent two-dimensional depth map.

The assumption of a static scene with all image plane motion being the result of camera motion is known as camera egomotion. In principle, for the more general case of a non-static scene, two separate views using two cameras in different known relative positions could be acquired to give a snapshot disparity or depth map. If a sequence of such snapshots were acquired then the evolving depth map for a non-static scene could be generated.

2.3.2 The Epipolar Constraint

The situation for a pair of cameras in general relative position and orientation is shown in figure 2.2. A three-dimensional point M projects to m_L in the left image plane R_L and to m_R in the right image plane R_R. All the points M that could project to m_L lie on the line joining m_L to the optical centre C_L and extending to infinity. Lines project to lines, so that the image of this line in the right image plane is a line. But this line in the right image is the locus of all possible points which correspond to the point m_L in the left image i.e. m_R must lie on the line which is called the epipolar

line. This constraint is called the epipolar constraint. The right hand epipolar line for a point M must pass through the image of C_L in R_L, and this point is called the epipole of R_R with respect to R_L. The epipolar constraint is symmetrical, so that an epipolar line also exists in R_L.

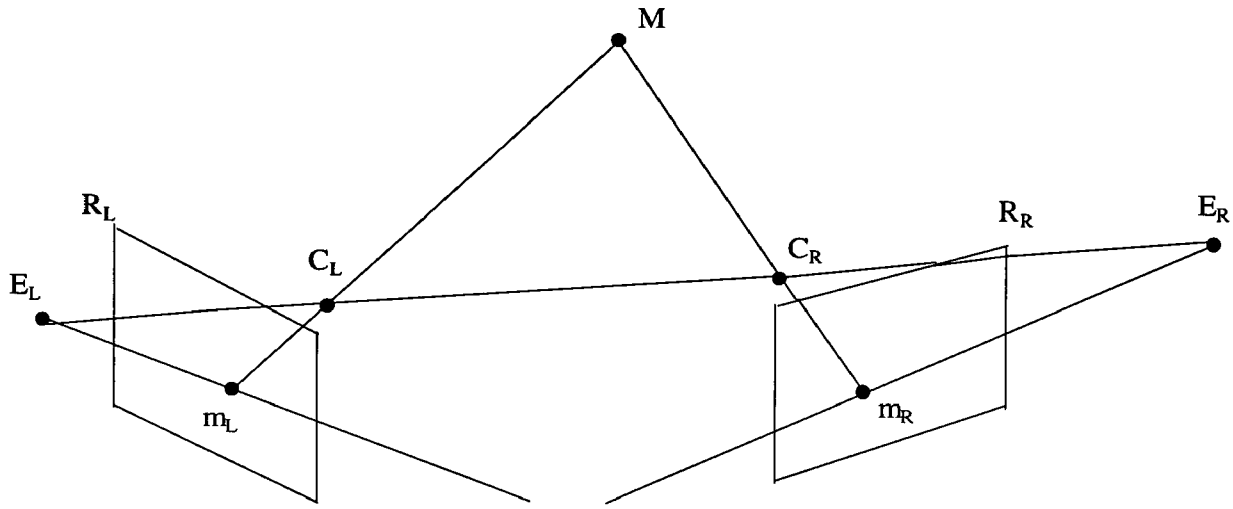

Figure 2.2: Epipolar geometry for a pair of cameras in general relative positioning and orientation

The epipolar lines lie on the intersection of the plane defined by M, C_L and C_R, called the epipolar plane, with the planes R_L and R_R . If the planes of R_L and R_R are arranged to be parallel to the line C_L-C_R then the epipolar lines in R_L and R_R are parallel. In particular if the conditions of section 2.3.1 are met such that equations 2.47 and 2.48 hold with camera motion along the x-axis only then the epipolar lines lie along the scanlines of the camera image detector array. This means that a search for corresponding points between two images need only be conducted along the scanlines.

If the camera motion is along the x-axis only, but there exists some error in mounting the camera or the image sensing array such the angles of pan φ and swing κ are non-zero then the image plane motion as the camera moves is given by:

$$\begin{pmatrix}\Delta U\\ \Delta V\\ \Delta S\end{pmatrix} = \begin{pmatrix}0 & 0 & 0 & \Delta t_x(\alpha_u \cos(\phi)\cos(\kappa) + u_0 \sin(\phi))\\ 0 & 0 & 0 & \Delta t_x(-\alpha_v \cos(\phi)\sin(\kappa) + v_0 \sin(\phi))\\ 0 & 0 & 0 & \Delta t_x \sin(\phi)\end{pmatrix}\begin{pmatrix}X\\ Y\\ Z\\ T\end{pmatrix} \qquad 2.49$$

In all the work with real images described in this thesis, the camera motion has been constrained to be along the x-axis only. The camera has also been mounted so that the camera image detector array plane coincides with the x-y plane of motion. However some small errors in swing and pan will still exist. From equation 2.49 an estimate of the effect of these small errors on the assumption that the image plane shift occurs along the detector scanline can be made. The change in u-axis position for pure x-axis translation of the camera under the epipolar constraint is given by equation 2.47. The corresponding movement along the v-axis is zero.

The equations 2.47 and 2.48 can be modified by dividing by the dimension of a sampled detector pixel in metres, p so that all measurements are in pixel units. If it is assumed also that the pixels are square so that $\alpha_u = \alpha_v = -f$. and that $p = 4 \times 10^{-5}$ m, z =1m, and f = 25 mm, which are typical figures for the camera used in the experiments described in the thesis, then from equation 2.47, Δt_x = 1.6 mm for one pixel shift along the u-axis. Assuming an error of 3 degrees in both swing and pan, equation 2.49 predicts that the true u-axis shift will be 1.007 pixels. The corresponding v-axis shift is -0.04 pixels. Under these conditions the assumption that it is only necessary to search along the image detector scanlines is weakened by approximately half a v-axis pixel in ten u-axis pixels.

2.3.3 Determining image plane motions

There are two main approaches to determining the image plane motions Δu and Δv in equations 2.47 and 2.48. These are the gradient-based or optic flow methods (Horn and Schunk, 1986) and

the correlation-based or matching methods. Both these methods share very similar underlying assumptions. In the work described in this thesis, the correlation-based approach has been used.

The Depth from optic flow method (*op cit.* Horn and Schunk, 1986) aims to compute an instantaneous velocity vector for points in an image, from sequences of images starting with the first image. It also assumes that local brightness or intensity patterns within an image do not change between images, but simply change their position within the image frame. Thus if the brightness of an image point (u,v) at time t is represented by $I(u,v,,t)$, then for a small time interval δt under this assumption:

$$I(u,v,t) = I(u+\frac{\partial u}{\partial t}\delta t, v+\frac{\partial v}{\partial t}\delta t, t+\delta t) \qquad 2.50$$

As the time interval δt tends to zero, then:

$$\frac{\partial I(u,v,t)}{\partial u}.\frac{\partial u}{\partial t} + \frac{\partial I(u,v,t)}{\partial v}.\frac{\partial v}{\partial t} + \frac{\partial I(u,v,t)}{\partial t} = 0 \qquad 2.51$$

Equation 2.51 allows the computation of the image velocities at a point u,v, given measurements of the spatial derivatives of the image intensity and the time derivative of the image intensity with the following restriction. From equation 2.51, it can be seen that along any line of constant brightness then the components of $\frac{\partial I}{\partial u}$ and $\frac{\partial I}{\partial v}$ along that line will be zero. Thus 2.51 does not allow a determination of image velocity components along lines of constant intensity, but only along lines perpendicular to the lines of constant brightness. This is called the aperture problem. Because of

the aperture problem equation 2.51 does not of itself allow a determination of the instantaneous image velocities, and an additional constraint is needed. In a method proposed by Horn and Schunk (*op cit.* Horn and Schunk, 1986) the additional constraint adopted is that the image velocities are smooth everywhere. The problem of determining the image velocities is then recast as a variational problem in which a functional E is minimised where:

$$E = \iint \left[|\nabla I| U\perp + \frac{\partial I}{\partial t} \right]^2 du\,dv \quad + \quad \lambda \iint \left| \nabla \left[\frac{\partial u}{\partial t} \right] \right|^2 + \left| \nabla \left[\frac{\partial v}{\partial t} \right] \right|^2 du\,dv \qquad 2.52$$

The first integral represents the constraint of equation 2.51 modified to be in terms of the component of image plane velocity perpendicular to the line of zero brightness gradient, and the second integral term represents a smoothness constraint on the image velocities. The constant λ balances the importance assigned to the smoothness constraint and the constant intensity constraint which is now effectively weakened to a minimal intensity change constraint. Solutions for $\frac{\partial u}{\partial t}$ and $\frac{\partial v}{\partial t}$ which minimise equation 2.52 are obtained by numerically solving the corresponding Euler equations. The optic flow technique requires that the image be sampled sufficiently quickly to avoid aliasing. The chief weaknesses of the optic flow method is that the velocities at points of depth discontinuities are discontinuous and therefore the smoothness constraint is invalid.

The correlation-based approach aims to directly determine the displacements of corresponding points in two images. The Depth from Stereo method for two images can be summarised in the following steps:

- Points in the two images that are both the perspective projections of a common three-dimensional point are identified. These image points are the corresponding points.

- The change (disparity) in the image co-ordinates of the image points is measured.

- The depth is recovered from the known relative positions and orientations of the camera viewpoints and the projection matrices.

As pointed out by (Marr, 1979), the main difficulty with the extraction of three-dimensional scene structure using the stereo approach is that of determining the correspondence of points. Correlation based methods attempt to establish correspondence of points by extracting a numerical measure of the difference between the brightness patterns in small patches centred on the point to be matched in the first image and the candidate matches in the second image. The assumption of (nearly) constant image brightness from image to image that is assumed in the gradient-based methods is also implicitly assumed in correlation-based methods. Some possible measures of match, measured over a matching patch A, between the brightness (or grey-scale) values of pixels $h(u, v)$ in one image and $g(u, v)$ in another image might be:

$$\max_A |h-g| \qquad 2.53$$

$$\iint_A |h-g|, \qquad 2.54$$

$$\iint_A (h-g)^2 \qquad 2.55$$

A normalised cross correlation can be derived (Rosenfeld and Kak, 1982) from 2.55 for discreetly sampled images, which is:

$$\text{corr coeff}_{\text{direct norm}} = \frac{\sum\sum g(i,j)h(i,j)}{\sum\sum g^2(i,j)\sum\sum h^2(i,j)} \leq 1 \qquad 2.56$$

This correlation measure is called the direct normalised cross correlation. If the grey-scales of the images h and g are normalised by subtracting the average grey level of the image from each pixel, a mean normalised match measure is generated:

$$\text{corr coeff}_{\text{mean norm}} = \frac{\sum\sum(h-\mu)(g-\nu)}{\sqrt{\sum\sum(h-\mu)^2\sum\sum(g-\nu)^2}} = \frac{\left(\left(\sum\sum\frac{hg}{A}\right)-\mu\nu\right)}{\sigma\tau} \qquad 2.57$$

Where μ and ν are the means and σ,τ the standard deviations of the images.

It is known that cross correlation of a patch in one image with candidate patches in another is the same as convolution with a filter that has the complex conjugate spatial frequency response to the spatial frequency response of the image patch to which a match is sought. This is exactly analogous to the matched filter from communication theory which maximises the signal to noise ratio under the assumption that the signal which is sought is corrupted by additive white noise. The matched filter, which has the complex conjugate frequency domain response to the image patch to be matched, has a space domain point spread function which is the same as that of the image being matched but rotated by 180^0 and shifted to the centre pixel of the candidate match patch. Because correlation is the same as 'flipped' convolution, convolution with such a point spread function is the same as correlation with the point spread function of the original image to be matched. This provides a signal to noise ratio justification for correlation as a matching measure.

As is pointed out by (*op cit* Rosenfeld and Kak, 1982), signal to noise ratio is not the only criterion that can be used to define optimality for a matched filter. Another criterion that is suggested is to maximise the ratio of the expected value of the filter output at the matching point to the variance of the filter output taken over all points. This criterion produces an optimum filter whose point spread function contains derivatives of the brightness values for search spaces that are locally correlated, and filters whose point spread function are the same as the matched filter for highly uncorrelated search spaces. In effect this is the same as using the outline of the object to be matched as a correlation template. This argument leads logically to feature-based matching which is robust and stable to noise but which produces sparse depth information. A study of correlation-based matchers of the type discussed above is presented in (Burt *et al.*, 1982).

A correlation-based matcher which implements equation 2.55 directly is the Sum of Squared Differences (SSD) matcher and this matcher is widely used in the literature. (*op cit.* Anandan, 1984), (Anandan, 1989), (*op cit.* Matthies *et al.*, 1989), (Corbatto *et al.*, 1995), and (Trucco *et al.*, 1996). In the work described in this thesis the SSD matcher is used as the matching mechanism. The theory and performance of the SSD matcher is left to be examined in detail in Chapter 3

2.3.4 Tracking of image plane motion through a sequence of images

Under the assumption of planar x-axis motion the shift along the u-axis of a point in the image plane is given by equation 2.47. From 2.47 it can be seen that for a point at some fixed depth the shift along the u-axis is proportional to the camera shift. For a fixed precision in disparity determination from the matcher, a larger camera shift gives better depth precision. However, the larger the camera shift, the greater is the range of possible candidate matches. This increases the chances of a false match, and therefore gross error in the depth measurement.

An approach due to Matthies *et al* (*op cit.* Matthies *et al.*, 1989), and which has been adopted by other workers (Distante *et al.*, 1992), (*op cit.* Corbatto *et al.*, 1995), and (*op cit.* Trucco *et al.*, 1996). is therefore to track either the depth or the disparity through a sequence of images where the camera shift is very small between images. At each matching step the search space is constrained to a small range, given some knowledge of the range of depths encountered in the visible scene, which reduces the possibility of false match. However, at the end of a sequence of images the camera shift between the first and last image can be large, giving the potential for good depth precision. A Kalman filter (*op cit.* Kalman, 1960) is then used to track the evolution of the disparity or depth and to update the depth map after each matching step. The Kalman filter relies on the matcher giving an estimate of the uncertainty in each measurement of disparity (see section 2.4). At each matching step there also exists a measure of the depth uncertainty at each pixel. The equations for the scalar Kalman filter for disparity tracking used in the work with image sequences described in this thesis are:

Process Model:

$$d_{[n+1]} = \frac{(n+1)}{n}.d_{[n]} + g_{[n+1]} \qquad 2.58$$

Where $d_{[n]}$ = disparity in pixels at camera step (image) n. $g_{[n]}$ is zero mean Gaussian noise with variance σ_g^2

Measurement Model:

$$M_{[n+1]} = C.d_{[n+1]} + v_{[n+1]} \qquad 2.59$$

$M_{[n]}$ = measured disparity. C = measurement constant = 1. $v_{[n]}$ is zero mean Gaussian noise of variance σ_v^2 due to the measurement process.

Estimator Equation:

$$\hat{d}_{[n+1]} = \frac{(n+1)}{n}.\hat{d}_{[n]} + K_{[n+1]}.[\, M_{[n+1]} - C.\frac{(n+1)}{n}.\hat{d}_{[n]} \,] \qquad 2.60$$

Where $\hat{d}_{[n]}$ = estimated disparity at step$[n]$;

Updated Mean square error:

$$P_{[n+1]} = \frac{1}{C}.K_{[n+1]}.\sigma^2_v \qquad 2.61$$

Filter Gain:

$$K_{[n+1]} = \frac{C\,[(\frac{n+1}{n})^2.P_{[n]} + \sigma_g^2\,]}{\sigma_v^2 + C^2.\sigma_g^2 + C^2.(\frac{n+1}{n})^2.P_{[n]}} \qquad 2.62$$

The disparity, $d_{[n]}$, for the step $n = 1$ is obtained after the first pair of images in a sequence are matched. The estimated measurement noise variance is supplied by the SSD matcher, and the mean squared error for each pixel being tracked is initialised to a high value.

2.4 Summary of Chapter 2

Chapter 2 has presented an overview of the stereo matching approach to depth recovery that is used in this thesis. The image formation process under the perspective projection has been examined and the requirements for camera calibration and the determination of disparity for depth recovery have been reviewed. The epipolar constraint and the effect of limiting camera motion to x-axis motion only have been discussed as simplifying assumptions for the problem of depth map determination. The two general approaches to measuring image plane motion have been outlined, and the use of correlation-based matchers in the stereo matching approach has been discussed. The SSD matcher has been introduced, and the use of the SSD matcher's potential to produce both sub-pixel estimates of disparity and estimates of the uncertainty in those estimates in a Kalman filter has also been discussed. The SSD matcher is used in this thesis to generate depth maps. Therefore, because the design of filters for depth map filtering, which is the main thrust of this thesis, depends in part on knowledge of the noise processes associated with the matcher, the SSD matcher is examined in detail in Chapter 3.

2.5 References

(Anandan, 1984) Anandan P., "Computing dense displacement fields with confidence measures in scenes containing occlusion. " Proceedings DARPA image understanding workshop. pp 236-246, 1984

(Anandan, 1989) Anandan, P. "A Computational Framework and an Algorithm for the measurement of visual motion." International Journal of Computer Vision. Vol 2 pp. 283-310. 1989.

(Burt *et al.*, 1982) Burt, P. J., Chihsung, Y., and Xinping, X. "Local Correlation Measure for Motion Analysis, a Comparative Study." Proceedings IEEE Conference on Pattern Recognition and Image Processing. pp. 269-274 1882.

(Corbatto *et al.*1995) Corbatto M. Tinonin S. Trucco E. and Roberto V. "Dense Depth Maps From Motion Using Dynamic Data Fusion". Proceedings of IEE Conference on Image Processing and its Applications. pp 642-646. July 1995.

(Distante *et al.*, 1992) Distante A., Lovergine F. P., Attolico G., Chiardia M. T., and Caponnetti L. "Stochastic Structure Estimation By Motion." Proceedings SPIE International Conference on Robots and Computer Vision XI pp 476-485. 1992.

(Faugeras, 1993) Faugeras O.D. "Three Dimensional Vision: A Geometric Viewpoint." MIT Press. 1993

(Haralick and Shapiro, 1993) Haralick R.M. and Shapiro L.G, "Computer and Robot Vision." Vol. 2. Addison-Wesley. 1993.

(Horn and Schunk, 1986) Horn, B. K. P and Schunk, B. G. "Determining Optical Flow." Artificial Intelligence Vol. 17 pp185-203. 1981.

(Horn, 1986) Horn B.K.P. "Robot Vision" MIT Press. 1986

(Kalman, 1960) Kalman, R.E. "A New Approach to Linear Filtering and Prediction Problems." ASME Journal of Basic Engineering pp 25-45 March 1960.

(Marr, 1979) Marr, D. "A computational theory of human stereo vision." Proceedings of the Royal Society of London B204 pp301-328. 1979.

(Matthies *et al.*, 1989) Matthies L., Kanade T.,and Szeliski R., "Kalman Filter-based Algorithms for Estimating Depth from Image Sequences." International Journal of Computer Vision. No. 3, pp 209-236, 1989.

(Rosenfeld and Kak, 1982) Rosenfeld A and Kak A C "Digital Picture Processing" Vol 2 1982.

(Trucco and Verri, 1998) Trucco E. and Verri A. "Introductory Techniques for 3-D Computer Vision." Prentice Hall 1998.

(Trucco *et al.*,1996) Trucco E, Roberto V, Tinonin S and Corbatto M. "SSD Disparity Estimation for Dynamic Stereo." Proceedings British Machine Vision Conference Vol. 1, pp 343-352. 1996.

Chapter 3 : Performance of the SSD matcher

3.1 Introduction

The analysis of the noise processes associated with the SSD sub-pixel matcher that is made in this chapter shows that the noise process associated with the SSD sub-pixel technique is a mixture of Gaussian and impulsive noise. It also shows that there exists a minimum backstop noise associated with the correlation-based matcher that is reasonably independent of the image statistics. Furthermore, it is established using examples that the measure of uncertainty in the disparity due to (Matthies *et al.*, 1989) can be unreliable. Therefore, the filtering produced by the Kalman filter as outlined in section 2.3.4 is ineffective and spatial filtering of the depth map is required to remove the impulsive noise component. The chapter finishes by reviewing the relationship between filtering and regularisation of ill-posed problems, which provides a motivation for the use of filters based on fuzzy inferencing systems that is the primary subject of this thesis.

3.2 The Sum of Squared Differences (SSD) sub-pixel matcher

3.2.1 Introduction to section

This section discusses some of the ideas of previous workers on the use of the Sum Squared Difference (SSD) matcher and its performance (Anandan, 1984), (Anandan, 1989) (*op. cit.* Matthies *et al.* 1989). These ideas have been used in this thesis in the algorithms that have been implemented to produce noisy depth maps, which have then been used to test the performance of fuzzy logic-based smoothing algorithms. Tests on the behaviour of the SSD matcher are presented in section 3.3.

3.2.2 The SSD Matcher as a sub-pixel disparity estimator

For the case of stereo matching under the epipolar constraint, the SSD matcher chooses the value of disparity $\hat{d}(u, v)$ which in the continuous case minimises:

$$SSD_c(\hat{d},u,v) = \iint_{win} [L(u+h, v+k) - R(u+h+\hat{d}(u,v), v+k]^2 \, dh \, dk \qquad 3.1$$

(Where $L(u, v)$ and $R(u, v)$ are the grey level values in the left and right images respectively.)

In the discrete case for e.g. a 3x3 window, the matcher chooses $\hat{d}(i, j)$ to minimise:

$$SSD_d = \sum_{h=-1}^{h=+1} \sum_{k=-1}^{k=+1} [L(i+h, j+k) - R(i+h+\hat{d}(i, j), j+k)]^2 \qquad 3.2$$

(Where (h, k) are indices defining the matching patches)

Equation 3.2 can be used to give an estimate of disparity at pixel (i, j). It should be noted that the disparity $d(i, j)$ can be a continuous quantity. However, in the case of a simple discrete matcher the values of disparity are quantised in whole pixels. The actual value of the SSD at the pixel i, j could be taken as a measure of confidence in the disparity obtained.

(*op. cit.* Matthies *et al.*. 1989) report an SSD matcher which not only gives a sub-pixel (continuous) estimate of disparity but also yields a measure of the uncertainty or variance in that estimate. Their argument, which is based on continuous quantities, goes as follows:

Starting from equation 3.1, it is assumed that the two images L and R are produced from some underlying true image F which is corrupted by uncorrelated Gaussian noise n of zero mean and variance σ_n^2 so that:

$$L(u,v) = F(u,v) + n_1(u,v) \tag{3.3}$$

and

$$R(u,v) = F(u - d(u,v),v) + n_2(u,v) \tag{3.4}$$

(Note *d(u, v)* is the true disparity map)

Equation 3.1 can now be rewritten as:

$$\begin{aligned} &SSD(\hat{d},u,v) = \\ &\iint_{win} [\, F(u+h,v+k) - F(u+\hat{d}(u,v) - d(u,v) + h,\ v+k) \quad + \\ &\qquad n_1(u+h,v+k) - n_2(u+\hat{d}(u,v) - d(u,v) + h, v+k)]^2\, dhdk \end{aligned} \tag{3.5}$$

Now assuming that $\hat{d}(u, v)$ - $d(u, v)$ is small, i.e. .the estimate $\hat{d}$ is close to the true disparity d a Taylor expansion can be made :

$$\begin{aligned} &F(u+\hat{d}(u,v) - d(u,v) + h, v+k) = \\ &\qquad F(u+h,v+k) + [\hat{d}(u,v) - d(u,v)]F'(u+h,v+k) + \ldots O(\hat{d} - d)^2 \end{aligned} \tag{3.6}$$

Where: $$F'(u+h,v+k) = \frac{\partial F}{\partial h}$$

From 3.6 in 3.5:

$$\begin{aligned} &SSD(\hat{d},u,v) = \\ &\quad \iint_{win} [\hat{d}(u,v) - d(u,v)]^2 [F'(u+h,v+k)]^2\, dhdk \\ &\qquad + \iint_{win} 2[n_1 - n_2].[\hat{d}(u,v) - d(u,v)].[F'(u+h,v+k)]\, dhdk \\ &\qquad\quad + \iint_{win} [n_1 - n_2]^2\, dhdk \end{aligned} \tag{3.7}$$

Equation (3.7) is a quadratic in $[\hat{d}(u, v) - d(u, v)]$ of the form

$$a(u, v)[\hat{d} - d]^2 + b(u, v))[\hat{d} - d] + c(u, v) \quad 3.8$$

Where:

$$a(u,v) = \iint_{win} F'(u+h, v+k)^2 \, dhdk \quad 3.9$$

$$b(u,v) = 2\iint_{win} [n_1 - n_2][F'(u+h, v+k)] \, dhdk \quad 3.10$$

$$c(u,v) = \iint_{win} [n_2 - n_1]^2 dhdk \quad 3.11$$

A sub-pixel estimate of disparity is obtained as follows. The SSD for a fixed patch size is evaluated for a range of possible disparity values in whole pixels. A quadratic is fitted to the integer disparity value giving lowest SSD and its two neighbours. These three points are in effect considered to be samples of the true underlying SSD error surface about the true minimum. The minimum point of this quadratic gives the sub-pixel estimate of disparity. In the paper of (*op. cit.* Matthies *et al.* 1989) each scan line in the pairs of grey-scale images is expanded by a factor of four using cubic interpolation. This is said to ".... improve precision..." as it gives a sub sub-pixel estimate of the disparity. A point not explicitly mentioned in the paper is that the interpolation process makes the resulting SSD error surface fit the quadratic ideal more closely as it makes the derivatives of the grey-scale finite. The main reason for the improvement in precision obtained due to expansion of the grey-scale image is discussed further in section 3.3.3. Expansion of the grey-scale images by interpolation is not used in the experiments described in this thesis as it also increases considerably the computational burden of computing the disparity.

The minimum point of the fitted quadratic, corresponding to the minimum SSD is given by:

$$2.a(u,v).[\hat{d}-d]+b(u,v)=0 \qquad 3.12$$

so that the estimate $\bar{d}$ which minimises the SSD is given by:

$$\bar{d}=d-\frac{b(u,v)}{2a(u,v)} \qquad 3.13$$

The variance of the $\frac{b(u,v)}{2a(u,v)}$ term represents the residual error on the estimate of disparity, $\bar{d}$.

The coefficient *a(u, v)* is a sum of the squares of the derivatives of the underlying intensity image. It will be higher when the underlying image is 'busier' i.e. in areas of the image where there is higher information content. In uniform areas of the image this coefficient will be zero. The confidence (or inverse variance) in the disparity estimate is proportional to this coefficient. The variance in the best estimate is given by:

$$\text{estimated variance}=\frac{\text{var}[b(u,v)]}{a^2(u,v)} = \frac{2.\sigma_n^2}{a(u,v)} \qquad 3.14$$

where σ_n^2 is the variance of the noise in the image. Substituting in 3.8 for $[\hat{d}-d]$ from 3.13 the residual SSD at the minimum of the quadratic is given by:

$$SSD_{residual}=\frac{3b^2(u,v)}{4a(u,v)}+c(u,v) \qquad 3.15$$

3.3 Investigations of matcher performance – Gaussian noise component

3.3.1 Introduction to section

This section investigates the performance of the SSD matcher under the idealised conditions of no grey-scale image noise and no distortion of the grey-scale images. Under these conditions, the theory of the sub-pixel matcher described in section 3.2.2 should result in zero noise in the disparity map. However, in practice noise does occur in the disparity map even under these ideal conditions.

The images below (figure 3.1) are three random dot images. The bottom two images are the same as the first image but with offsets of 15 and 32 added to the grey-scale values of different areas of the image to create a step edge. The auto SSD vectors centred on column 5 of these three images are then plotted and shown in figure 3.2.

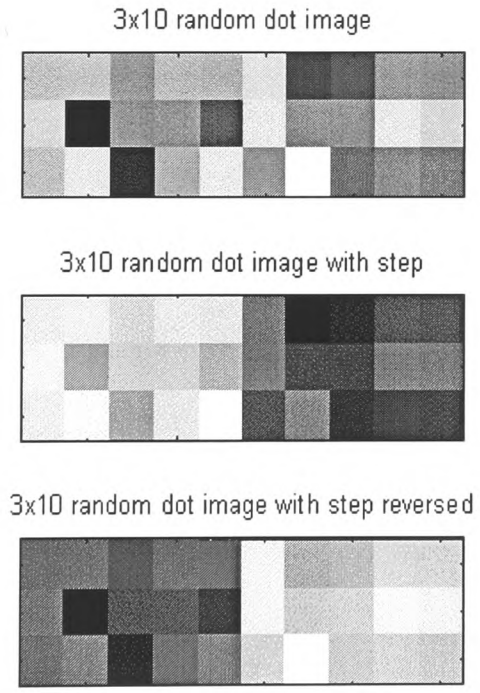

Figure 3.1: Three random dot images used for SSD matcher tests

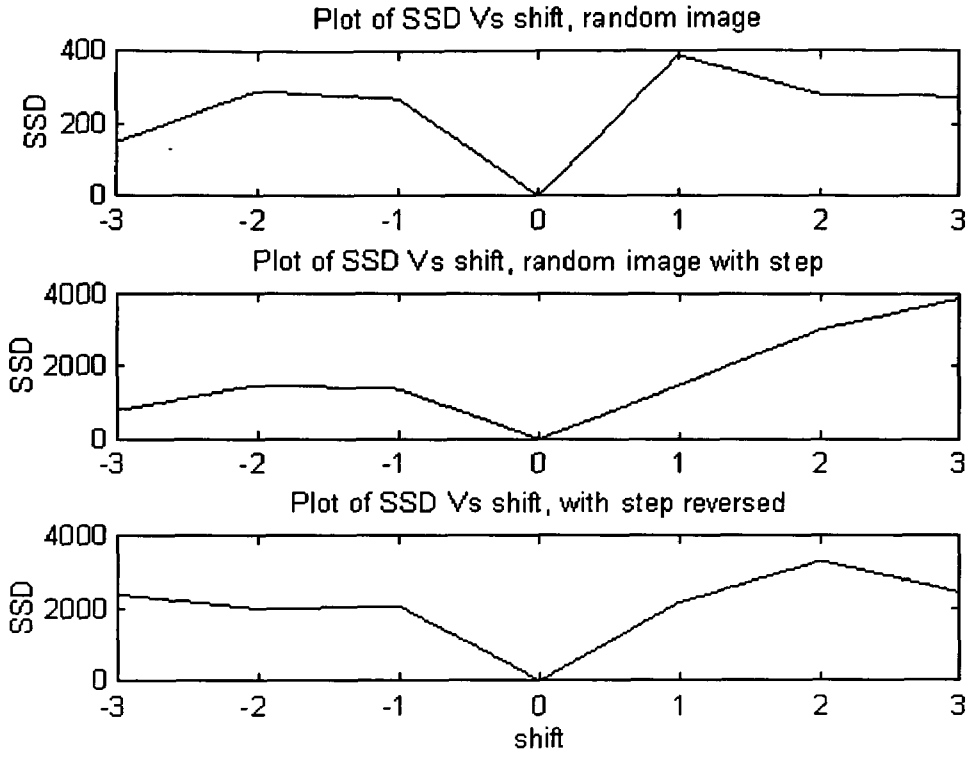

Figure 3.2: Plots of SSD versus shift for the random dot images of figure 3.1

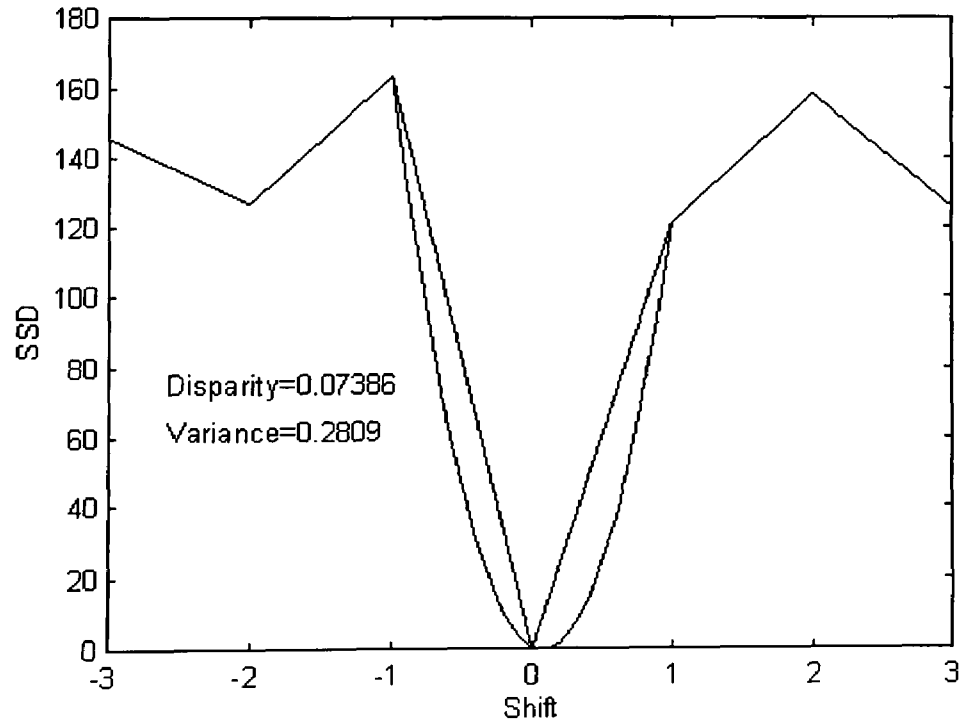

Figure 3.3: Example of a quadratic fitted to an SSD error surface

Figure 3.3 shows an example of a quadratic fitted to an SSD error surface. Because of the asymmetry of the SSD plot on either side of the true minimum the quadratic sub-pixel estimate is offset from the true value, which in this case equals zero pixels.

In order to test the 3x3 SSD matcher under ideal conditions the auto SSD of a large (64x64) random dot grey-scale image was calculated for each pixel (true disparity = 0) and a histogram plotted of the estimated disparities. The random dot image consisted of uniformly distributed pixel values of between 0 and 63 (It was generated using the MATLAB '63*Rand(64,64)' function). The resulting histogram is shown in figure 3.4.

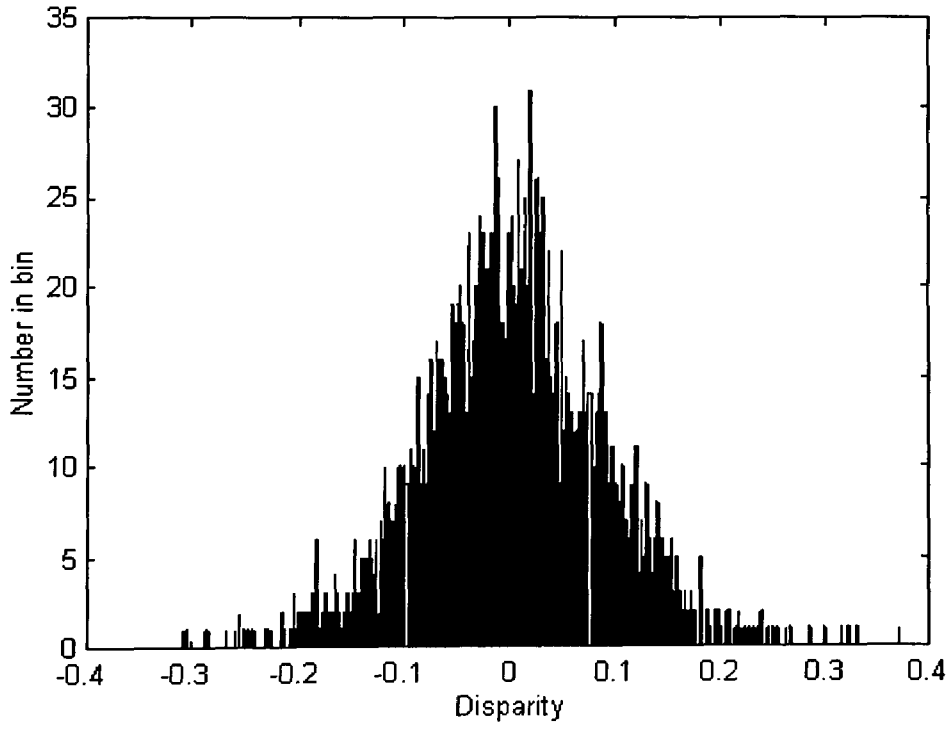

Figure 3.4: Histogram of disparities for image pair with no added noise

The distribution of errors is in this case of a Gaussian form. The theory described in section 3.2.2 does not predict such a Gaussian 'backstop' noise in the absence of additive noise or perspective

and camera distortions in the image forming process. However (Corbatto *et al.*, 1995), followed up in more detail in (Trucco *et al.*,1996), point out the existence of this backstop noise, which is a result of the quadratic interpolation scheme used to obtain the sub-pixel estimate. An alternative interpolation scheme based on heuristic assumptions is proposed in (*op cit* Trucco *et al.*,1996). Section 3.3.2 provides an explanation of the backstop Gaussian noise process observed in the quadratic interpolation case.

3.3.2 Analysis of Gaussian backstop noise

Consider the parabola fitted to the SSD surface generated by the SSD matcher as shown in figure 3.5. The position of the parabola minimum will vary partly as a result of true sub-pixel shifts under the assumptions of (*op cit* Matthies *et al.*, 1989) and partly as a result of variations due to the statistical distribution of the SSD values on either side of the minimum. The analysis described in section 3.2.2 neglects the terms in the Taylor expansion containing higher order derivatives and these are the terms that describe the variations in the SSD values on either side of the minimum. This section provides an explanation of these variations based on an analysis of the effect of the grey-scale image statistics on the statistics of the SSD values and hence on the position of the minimum of the quadratic used to obtain the sub-pixel estimate of disparity. The analysis assumes a simple random uniform distribution of grey-scale values and that the grey scale values are independent, but nevertheless provides an insight into the disparity map noise that occurs with more complex grey-scale image statistics.

In the case shown below in figure 3.5, the true SSD minimum is at a disparity of zero pixels. In order to analyse the noise on the disparity measurement due to the distribution of the adjacent SSD values it is necessary firstly to analyse the effect of the SSD values S_1, S_2, S_3 at the integer disparity positions d_0-1, d_0, d_0+1, on the position of the fitted quadratic minimum.

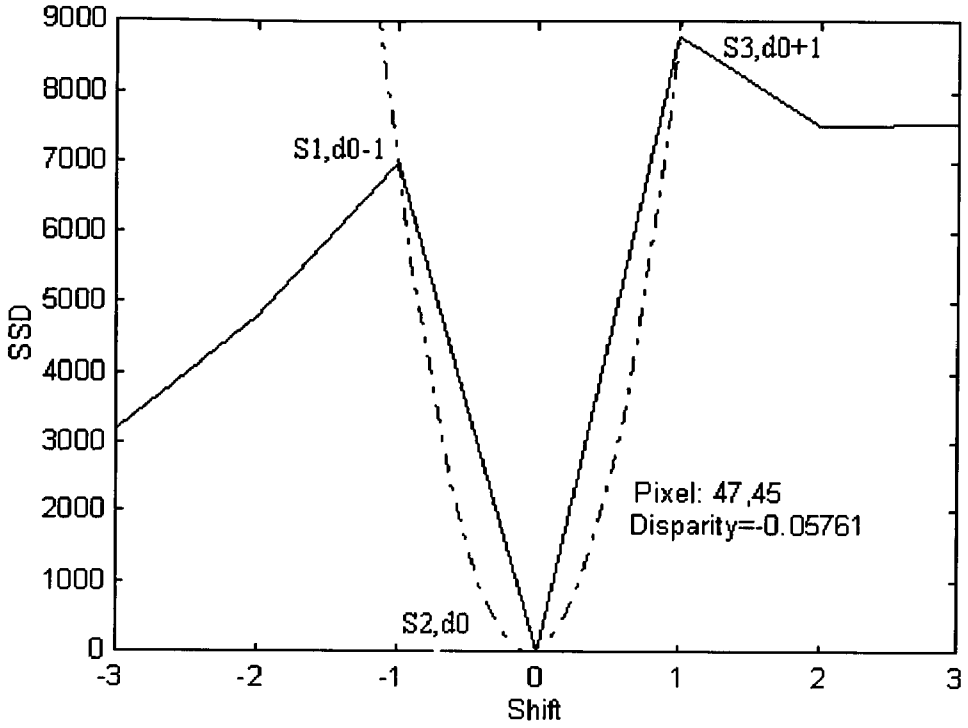

Figure 3.5: Illustration of the effect of SSD variations about the minimum SSD value on the sub-pixel measure of shift using the method of Matthies *et al.*

The fitted quadratic is given by Lagrange's formula (Davis, 1963) for the interpolating quadratic, S=Q(d) through three points:

$$Q(d) = \frac{(d-d_0)(d-d_0-1)}{(d_0-1-d_0)(d_0-1-d_0-1)} S_1 + \frac{(d-d_0+1)(d-d_0-1)}{(d_0-1-d_0)(d_0-1-d_0-1)} S_2 + \frac{(d-d_0+1)(d-d_0)}{(d_0-1-d_0)(d_0-1-d_0-1)} S_3 \qquad 3.16$$

After a rearrangement:

$$Q(d) = \frac{S_1+S_2+S_3}{2} d^2 - \frac{[2d_0(S_1+S_2+S_3)+1(S_1-S_3)]}{2} d + S_1(d_0(1+d_0)) + S_2 d_0^2 + \frac{d_0(d_0-1)}{2} S_3 + S_2 \qquad 3.17$$

The value of d which gives the minimum of this quadratic, $d_{min\,pos}$ is:

$$d_{min\,pos} = d_0 + \frac{S_1 - S_3}{2(S_1 + S_2 + S_3)} \qquad 3.18$$

It is assumed, without loss of generality, that $d_0=0$. In the absence of noise and perspective distortion in the original grey-scale images S_2 can be set to zero. Equation 3.18 becomes:

$$d_{min\,pos} = d_0 + \frac{S_1 - S_3}{2(S_1 + S_3)} \qquad 3.19$$

In order to determine the distribution of d_{minpos} it is necessary to determine the distributions of the sum-squared differences, S_1 and S_2. The simplifying assumption is made that the pixel grey-scale values are independent and uniformly distributed between 0 and 63 i.e. a uniform discrete joint probability distribution $P_f(R_1,R_2)$. The probability distribution P(SD) for the square of the difference, SD of any two pixels, R_1 and R_2 is given by:

$$P(SD) = \sum_{R_1,R_2 \in F} P_f(R_1,R_2) \qquad 3.20$$

Where F is the set of points R_1, R_2 such that $SD = (R_1 - R_2)^2$

Implementing equation 3.20 using MATLAB the probability distribution of the squared difference of two pixels is obtained and shown in figure 3.6 below. The distribution is discontinuous because the grey level values are quantised into 64 possible levels.

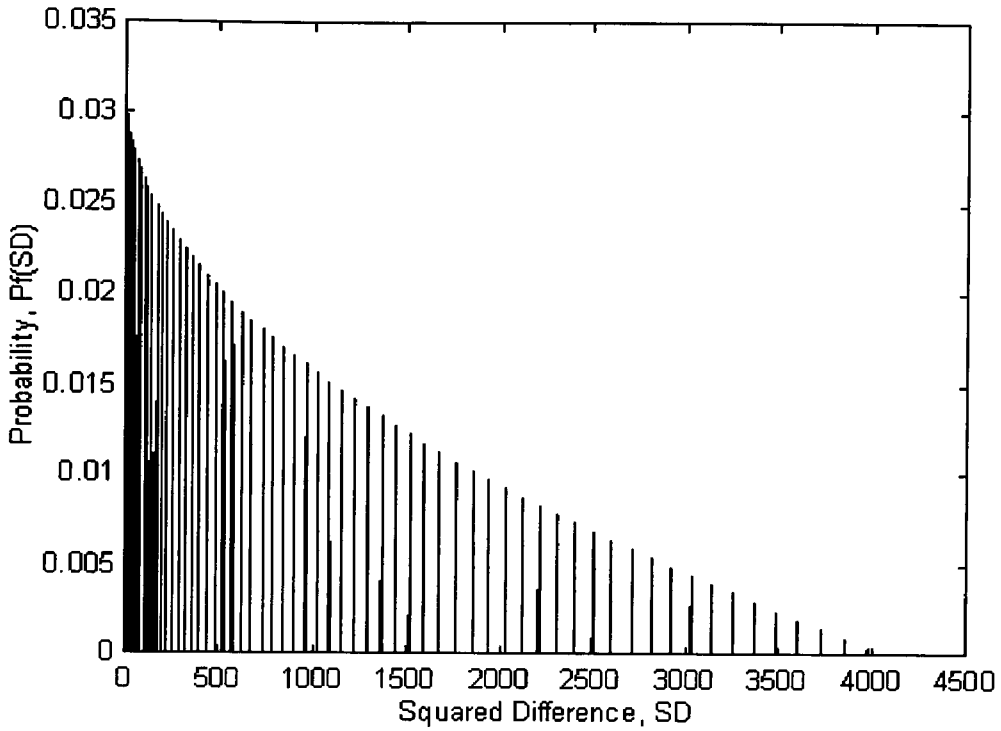

Figure 3.6: Probability distribution of the squared difference of two pixels with grey-scale values that are independent and uniformly distributed between 0 and 63

The probability distribution of the sum of two independent random variables each having probability distribution f_1 and f_2 is given by the convolution of f_1 and f_2 (Kendall and Stuart 1963). The probability distribution of the Sum of Squared differences (SSD) over a 3x3 patch, S_n is therefore given by the convolution with itself of the probability distribution shown in figure 3.6 carried out nine times. The result of doing this is illustrated below in figure 3.7. It can be seen that the distribution of SSDs in this case tends towards a Gaussian distribution as would be expected from the central limit theorem. The thickness of the plotted line is due to the discontinuities of the plot of figure 3.6. After discrete convolution nine times, the number of data points is very large and therefore in order to implement the next stages of the calculation in MATLAB the probability distribution of the SSDs was sub-sampled and approximated as a continuous distribution. This sub-sampled probability distribution is shown in figure 3.8

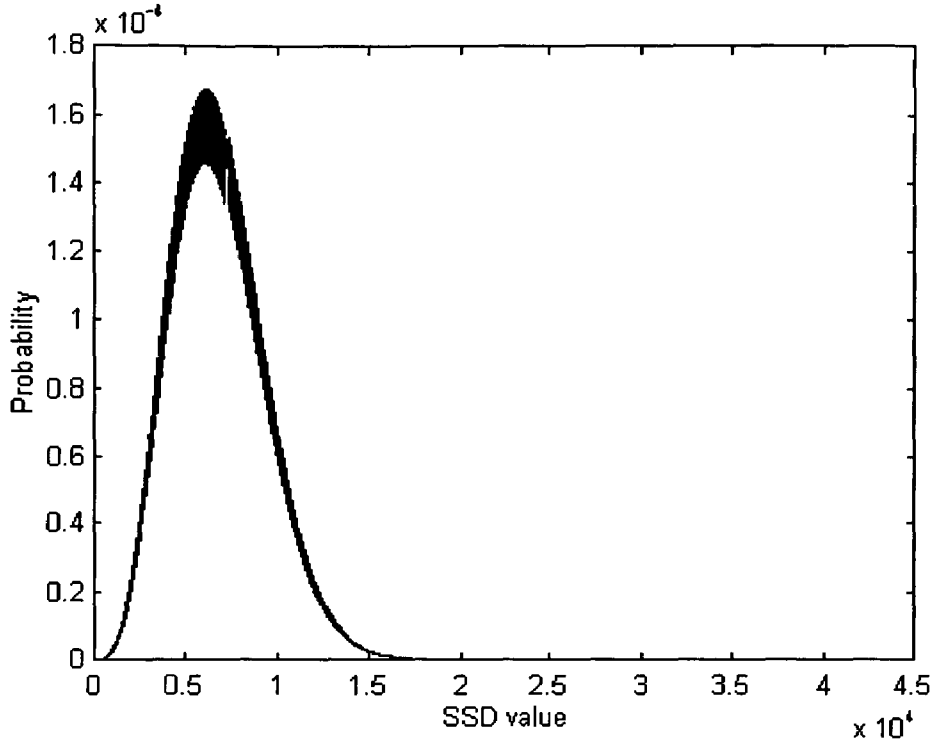

Figure 3.7: Probability distribution of the Sum of Squared differences (SSD) over a 3x3 patch.

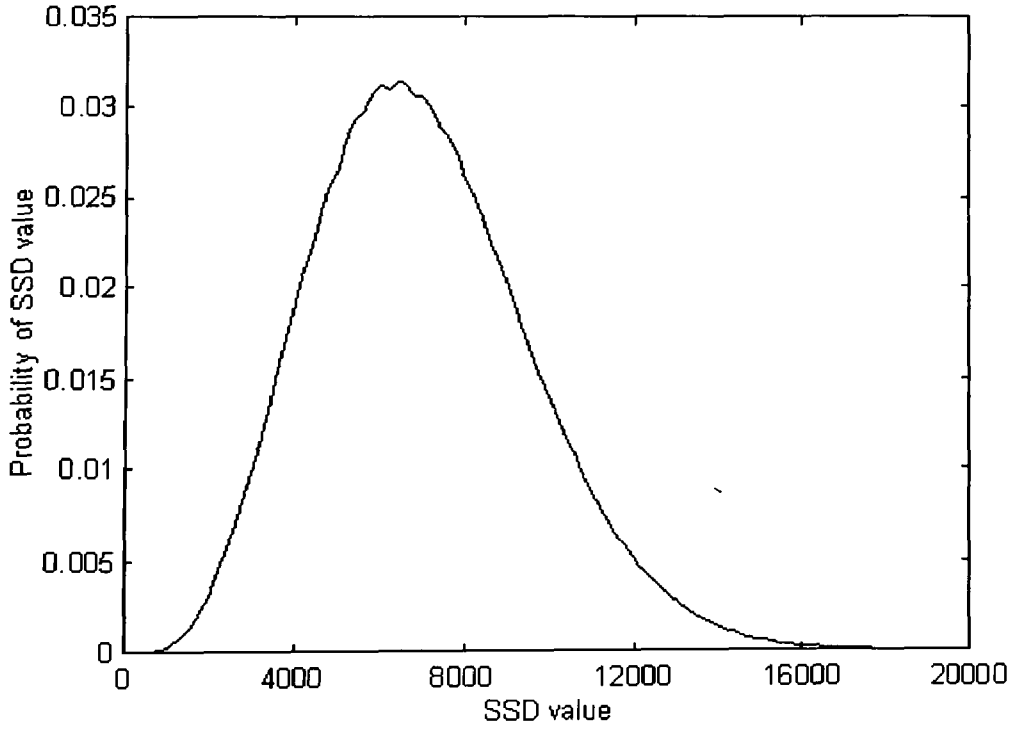

Figure 3.8: Sub-sampled probability distribution of the Sum of Squared differences (SSD) over a 3x3 patch

From equation 3.19, the sub-pixel disparity measure is given by:

$$d_{\min pos} = d_0 + \frac{S_1 - S_3}{2(S_1 + S_3)} \qquad 3.21$$

Which can be written as

$$d_{\min pos} = d_0 + \frac{Diff}{2(Sum)} \qquad 3.22$$

In order to find the probability distribution of the sub-pixel disparity it is necessary to find the probability distribution of the quotient:

$$\frac{S_1 - S_3}{2(S_1 + S_3)} = \frac{Diff}{2.Sum} \qquad 3.23$$

This can be done in three stages. Firstly the joint Probability distribution of 'Sum' and 'Diff' is determined. Then the joint probability distribution of the Quotient, $\frac{Diff}{2.Sum}$, is determined before finally determining the marginal probability distribution of the quotient alone. An assumption made as an approximation is that the SSDs S_1 and S_3 making up the sum and difference are independent random variables. This is not strictly true as the matching windows that generate S_1 and S_3 overlap. For 3 x 3 windows the overlap area is a minority of the total area and the independence assumption is reasonable as a first approximation. However, the assumption of independence of S_1 and S_3 is increasingly violated for larger matching windows. This increasing overlap area is illustrated in figure 3.9 for a 3 x 3 and a 5 x 5 matching window.

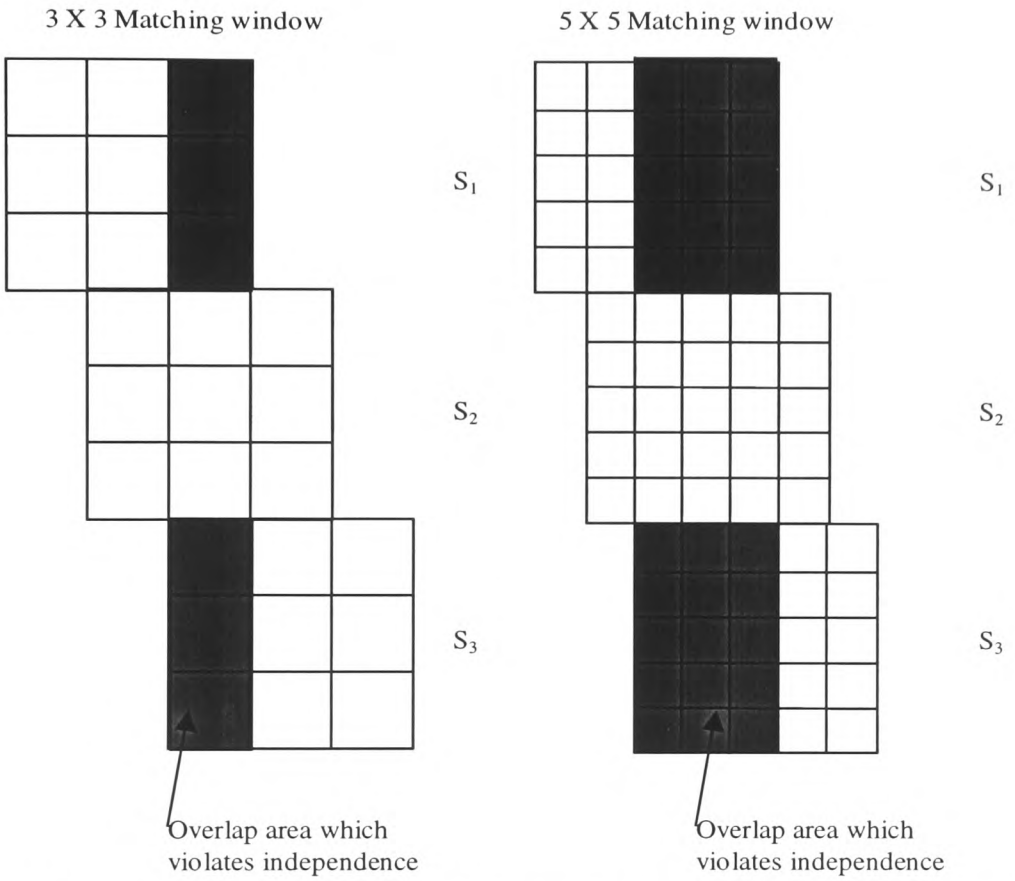

Figure 3.9: Illustration of increasing overlap area with matching window size, which causes violation of assumption of independence of SSDs S_1 and S_2

Under the independence assumption, which is reasonable for 3 x 3 matching windows:

$$Sum = S_1 + S_3 \tag{3.24}$$

$$Diff = S_1 - S_3 \tag{3.25}$$

So that:

$$S_1 = g_1(Sum, Diff) = 0.5.Sum + 0.5Diff \tag{3.26}$$

$$S_2 = g_2(Sum, Diff) = 0.5.Sum - 0.5Diff \tag{3.27}$$

The joint probability, P_{jsd} of random variables such as *Sum* and *Diff* that are related by equations of the form of 3.26 and 3.27 to random variables such as S_1 and S_3 is given by (*op cit.* Kendall and Stuart 1963):

$$\begin{aligned} P_{jsd}(Sum, Diff) &= P_{jss}(S_1, S_3).\left|J_{gg}\right| = P_s(S_1)P_s(S_3).\left|J_{gg}\right| \\ &= P_s(0.5.Sum + 0.5Diff).P_s(0.5.Sum - 0.5.Diff)\left|J_{gg}\right| \end{aligned} \quad 3.28$$

where $Pj(.)$ and $P_s(.)$ are the joint and marginal probability distributions and J_{gg} is the Jacobian:

$$J_{gg} = \det\begin{pmatrix} \frac{\partial g_1}{\partial \text{Sum}} & \frac{\partial g_1}{\partial \text{Diff}} \\ \frac{\partial g_2}{\partial \text{Sum}} & \frac{\partial g_2}{\partial \text{Diff}} \end{pmatrix} = -0.5 \quad 3.29$$

The joint probability distribution of *Sum* and *Diff* was calculated by implementing equations 3.28 and 3.29 in MATLAB and using the previously calculated probability distributions for the SSDs $P_s(S_n)$. This joint probability distribution is shown in figure 3.10.

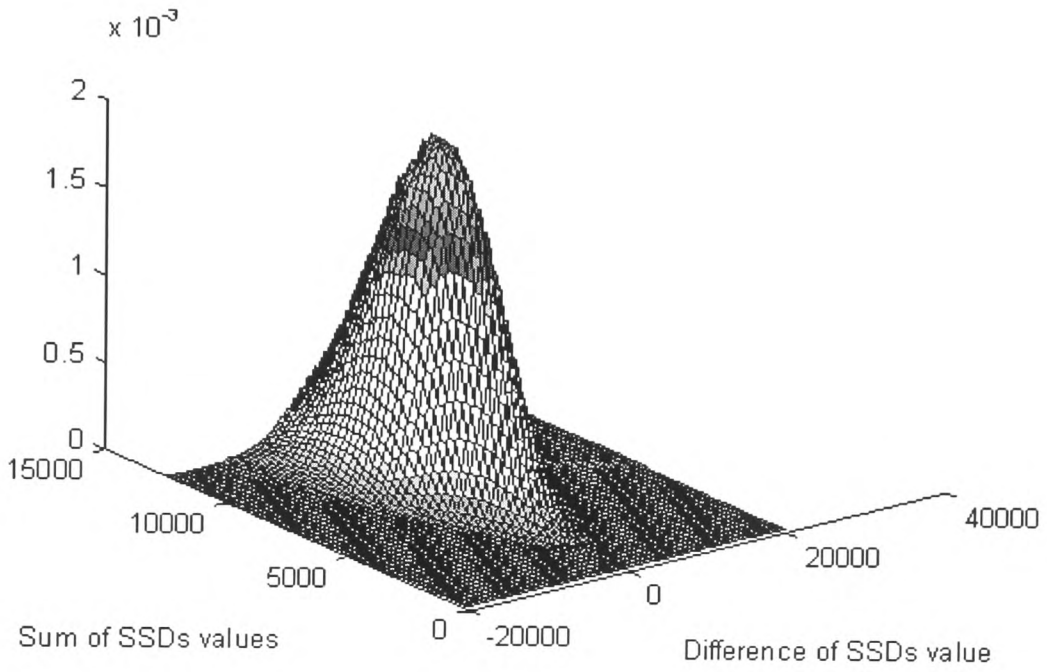

Figure 3.10: Joint probability distribution of sum and difference of SSDs

Now all that remains is to derive the marginal probability distribution of the quotient

$\frac{\text{Diff}}{2(\text{Sum})}$.

Define two new random variables:

$$Y_1 = \frac{Diff}{2Sum} \qquad 3.30$$

$$Y_2 = Sum \qquad 3.29$$

So that:

$$Diff = 2Y_1Y_2 = h_1(Y_1, Y_2) \qquad 3.30$$

$$Sum = Y_2 = h_2(Y_1, Y_2) \qquad 3.31$$

The joint Probability distribution of Y_1 and Y_2 ,$P_{jyy}(Y_1,Y_2)$, given by:

$$P_{jyy}(Y_1,Y_2) = P_{jsd}(2.Y_1Y_2,Y_2)|J_{hh}| \quad = 2Y_2.P_{jsd}(2.Y_1Y_2,Y_2) \tag{3.32}$$

Where the Jacobian J_{hh} is:

$$J_{hh} = \det\begin{pmatrix} \frac{\partial h_1}{\partial Y_1} & \frac{\partial h_1}{\partial Y_2} \\ \frac{\partial h_2}{\partial Y_1} & \frac{\partial h_2}{\partial Y_2} \end{pmatrix} = 2Y_2 \tag{3.33}$$

And P_{jsd} is determined using equation 3.28.

The joint probability distribution P_{jyy} is shown in figure 3.11. The marginal probability distribution of the quotient, P_{my}, which is the probability distribution of the sub-pixel disparities is given by:

$$P_{my}(Y_1) = \sum_{Y_2=-\infty}^{Y_2=+\infty} P_{jyy}(Y_1,Y_2) \tag{3.34}$$

This marginal probability distribution is shown in figure 3.12. Figures 3.13 and 3.14 show an actual histogram of sub-pixel disparities for a random dot image and the predicted distribution under the above assumptions plotted to the same disparity scale for comparison. It can be seen that even with the approximations and assumptions made in deriving the predicted distribution that the agreement between the prediction and the measured distribution is reasonably close, with a similar spread in values.

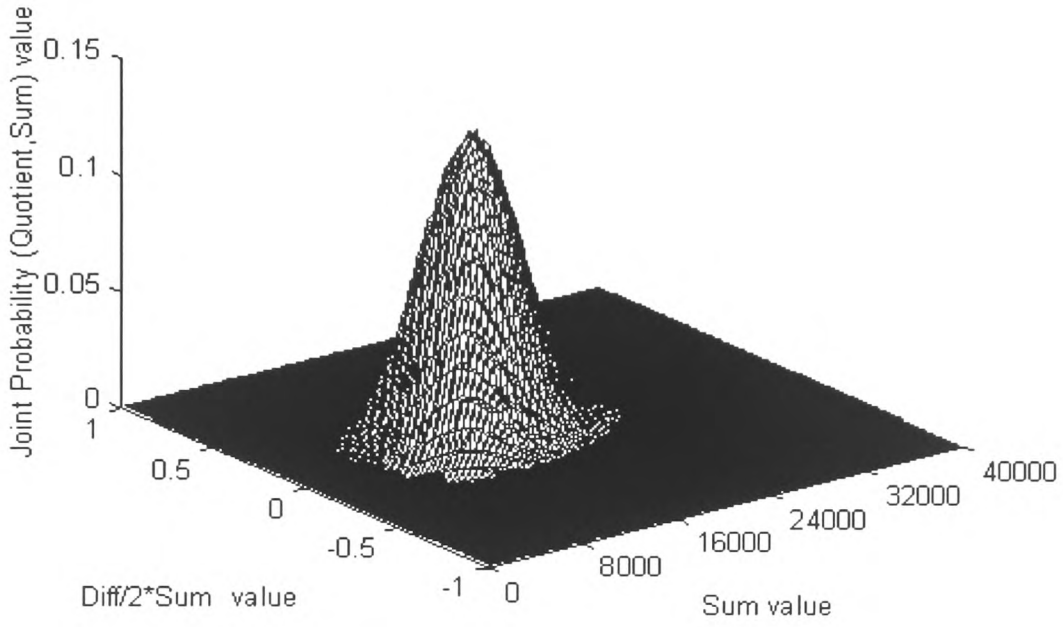

Figure 3.11: Joint probability distribution of quotient $\frac{Diff}{2.Sum}$ and sum of SSDs

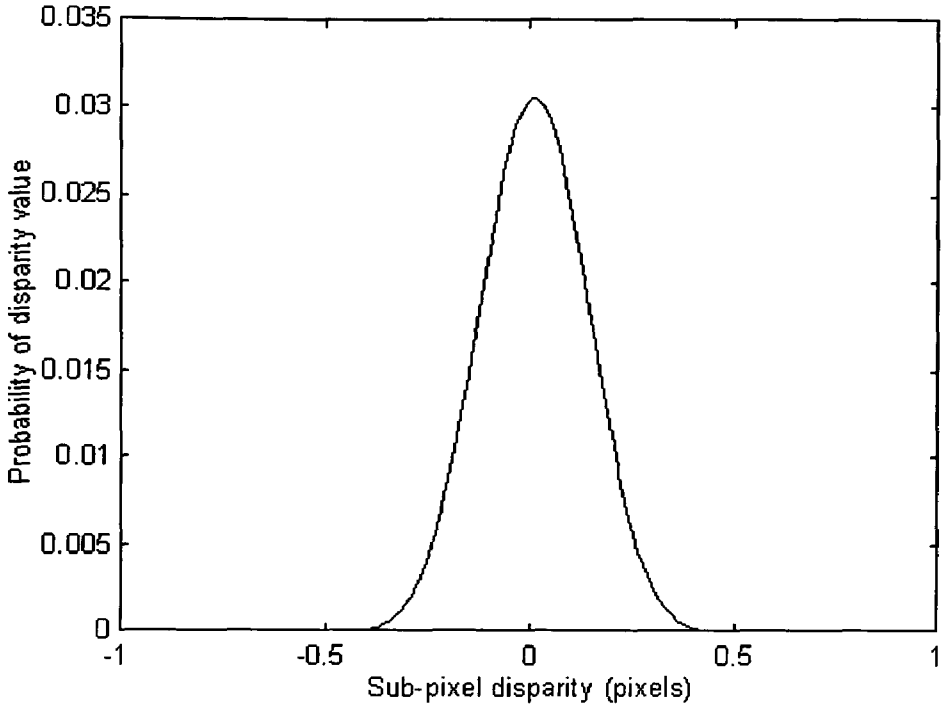

Figure 3.12: Predicted probability distribution of sub-pixel disparities for a uniformly distributed random dot image

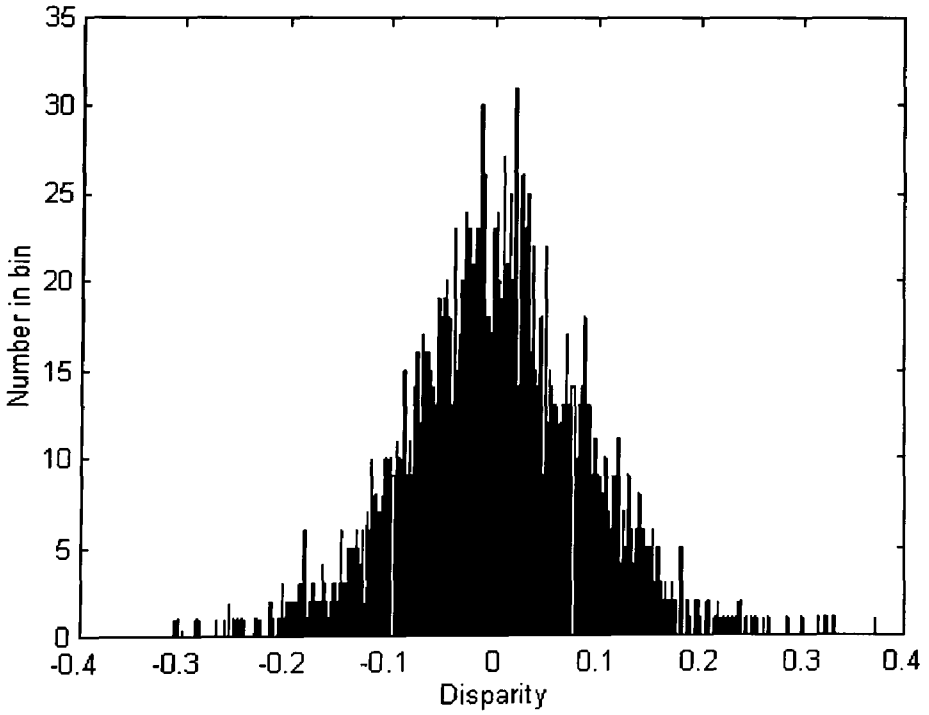

Figure 3.13: Histogram of sub-pixel disparities for a uniformly distributed random dot image

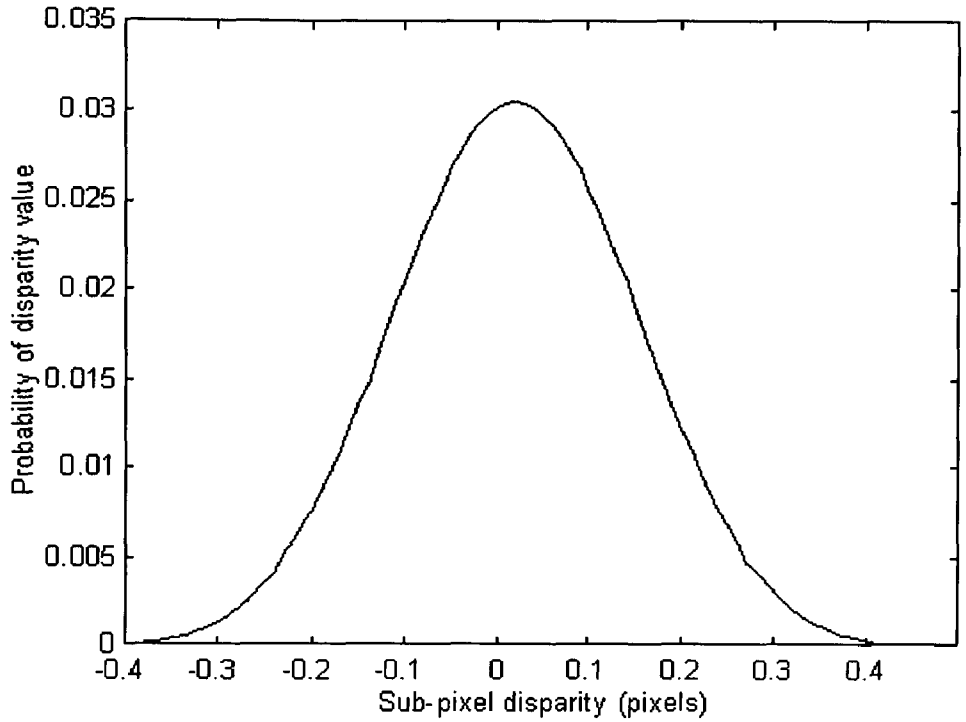

Figure 3.14: Predicted distribution of sub-pixel disparities for a uniformly distributed random dot image

Figures 3.15 and 3.16 show the actual and predicted distribution of disparities for a 5 x 5 matching window. In this case the prediction under the same assumptions as above does not agree as well with the actual measured distribution of disparities. However the prediction of a narrower distribution for a larger window is correct. The fact that the predicted distribution for the 5 x 5 window is wider than the measured distribution is due to the violation of the assumption that the SSDs on either side of the parabola minimum are independent. Section 3.3.3 examines the way that the width of the backstop distribution of disparities depends on the distribution of brightness values in the original images being matched.

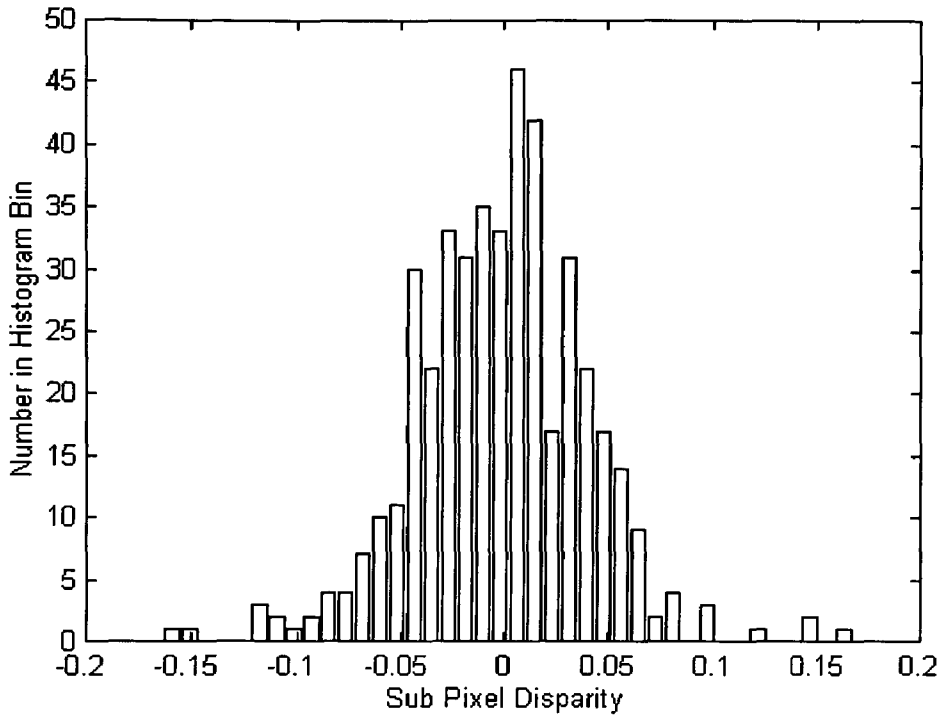

Figure 3.17: Actual distribution of disparities for a uniformly distributed random dot image using a 5 x 5 matching window

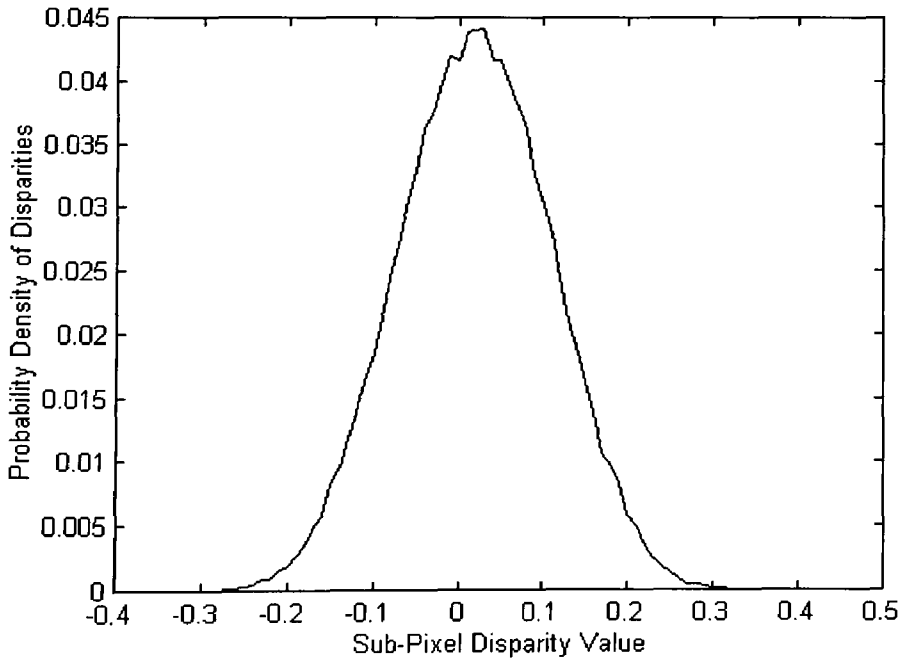

Figure 3.18: Predicted distribution of disparities for a uniformly distributed random dot image using a 5 x 5 matching window

3.3.3 Effect of grey scale image statistics on the distribution of disparity values

The approach used in section 3.3.2 to predict the distribution of disparities for a uniformly distributed grey scale image can be used to predict the distribution of disparities for grey scale distributions that are not uniform. The predicted and measured distributions of disparities for a Gaussian distributed grey scale image with a standard deviation of 5 and matched using a 3 x 3 window are shown in figures 3.19 and 3.20. For the cases of independent grey scale pixel values considered here, the distribution of disparities is found to be substantially independent of the statistics of the grey scale image in the absence of noise. However, the width of the disparity distribution depends on the matching window size. Larger matching windows give rise to narrower disparity distributions.

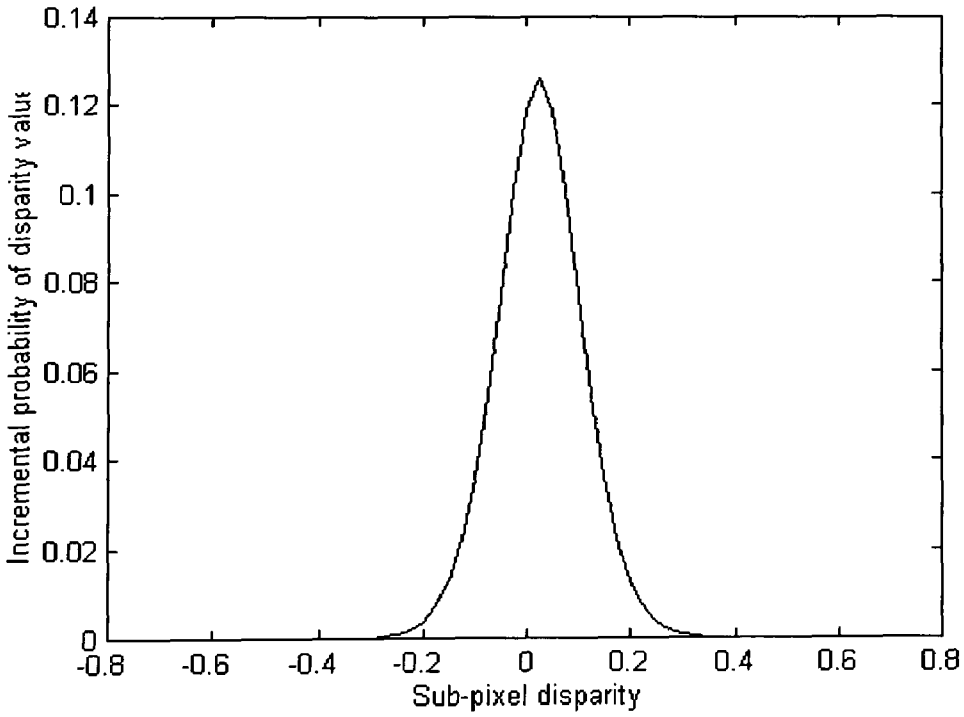

Figure 3.19. The predicted distribution of disparities for a 3 x 3 matching window and a Gaussian distributed grey scale image with a mean of 32 and standard deviation of 5

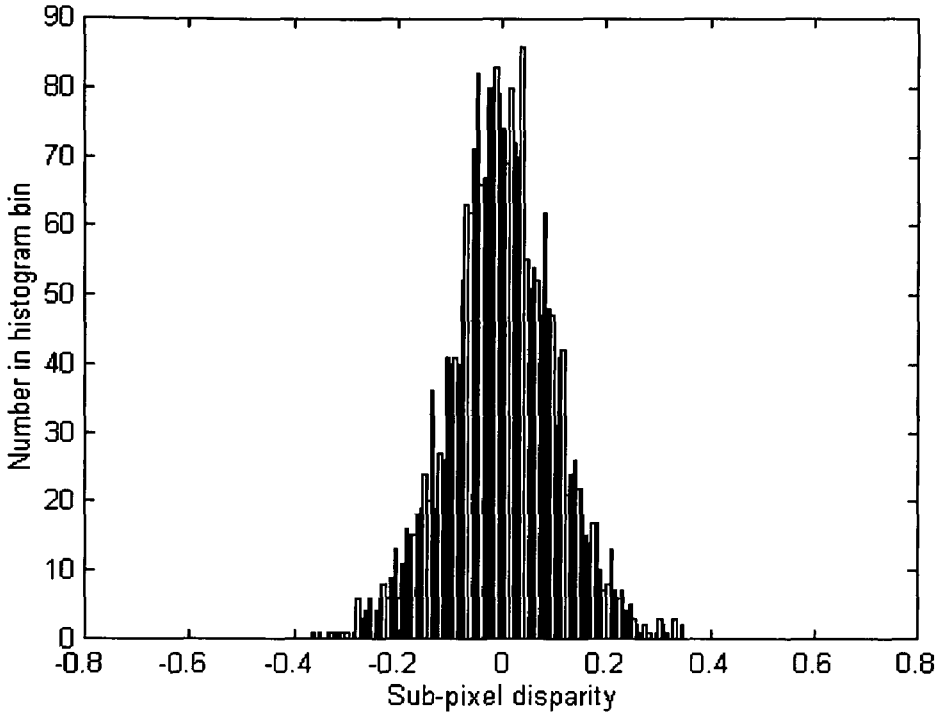

Figure 3.20. The measured histogram of disparities for a 3 x 3 matching window and a Gaussian distributed grey scale image with a mean of 32 and standard deviation of 5

The width of the distribution of disparities, and hence the precision in the disparity measurement for a fixed window size, is a fixed fraction of the smallest jump in disparity presented before fitting the quadratic to the error surface. Thus in the example shown in figure 3.20, the standard deviation of the disparities is 0.1 of a whole pixel. In the histogram of disparities shown in figure 3.21, the grey scale image has been expanded by cubic interpolation before matching, as was done by Matthies *et al.* (*op. cit.* Matthies *et al.* 1989). Although the shape of the histogram is no longer Gaussian in form, the width of the histogram measured by the extent of its tails and indeed the estimated standard deviation, as a fraction of the smallest jump in disparity is the same as that of figure 3.20. However this smallest jump is now only a quarter of a pixel. This is why grey-scale image expansion is successful in improving precision, as it reduces the effect of the backstop noise introduced by the quadratic method of determining sub-pixel disparity. However, this grey-scale

expansion considerably increases the computational load needed to compute a match. The reason for the interesting tri-modal shape of the disparity histogram, resulting from the interpolation between independent Gaussian distributed grey scale values is not pursued here.

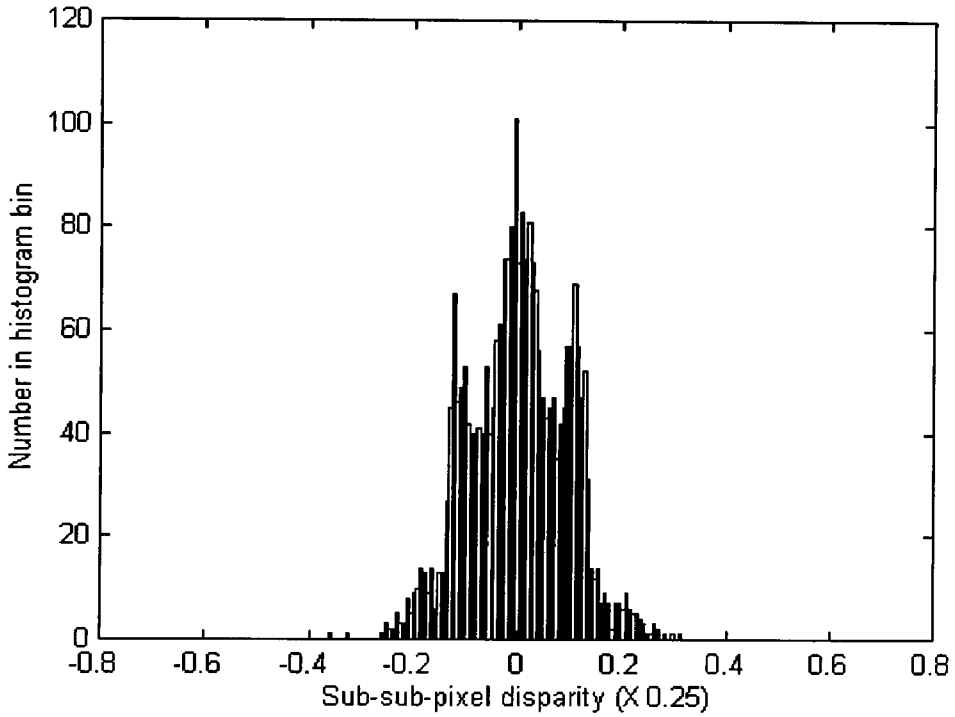

Figure 3.21. The measured histogram of disparities for a 3 x 3 matching window and a Gaussian distributed grey scale image with a mean of 32, standard deviation of 5, and expanded by a factor of four using cubic interpolation.

3.3.4 Transformation of disparity map noise to depth map noise

The discussion of noise above is confined to the noise in disparity maps. The distribution of noise in the corresponding depth map is corrupted by a transformed version of this noise. From equation 2.47:

$$z = \frac{f.\Delta t_x}{p\Delta u} \qquad 3.35$$

If the probability distribution function of the disparities is $P_D(\Delta u)$, the distribution of the depths is given by (*op cit.* Kendall and Stuart 1963):

$$P_z(z) = P_D\left(\frac{f.\Delta t_x}{p.z}\right).\left|\frac{d}{dz}\left(\frac{f.\Delta t_x}{p.z}\right)\right| = P_D\left(\frac{f.\Delta t_x}{p.z}\right).\frac{f.\Delta t_x}{p.z^2} \qquad 3.36$$

For a mean μ Gaussian distribution of disparity the distribution of depths becomes:

$$P_z(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{f.\Delta t_x}{\sqrt{2}.p.z\sigma} - \frac{u}{\sqrt{2}\sigma}\right)^2} .\frac{f.\Delta t_x}{p.z^2} \qquad 3.37$$

This distribution is illustrated in figure 3.22 for a camera with f = 25mm, p = 4 x 10^{-5} m, a camera translation Δt_x = 4 mm, and a mean disparity of 2.5 pixels with standard deviation 0.1 pixels (cf. Figures 2.15 and 2.16). The distribution of 3.22 is skewed, although for the particular camera parameter values listed above the distribution is still Gaussian in character (thin tailed and reasonably symmetric). However for objects at 2 m depth and therefore a mean disparity of 1 pixel, the same standard deviation of disparity values results in the depth distribution illustrated in figure 3.23. The same Gaussian distribution of disparities now results in a very broad and skewed distribution of depths. This illustrates the importance of minimising the Gaussian component of error in disparity maps, and that depth map noise is not Gaussian distributed, even when the disparity map noise is Gaussian. The shape of the depth map noise distribution is also strongly dependent on the camera parameters and is a function of the depth values.

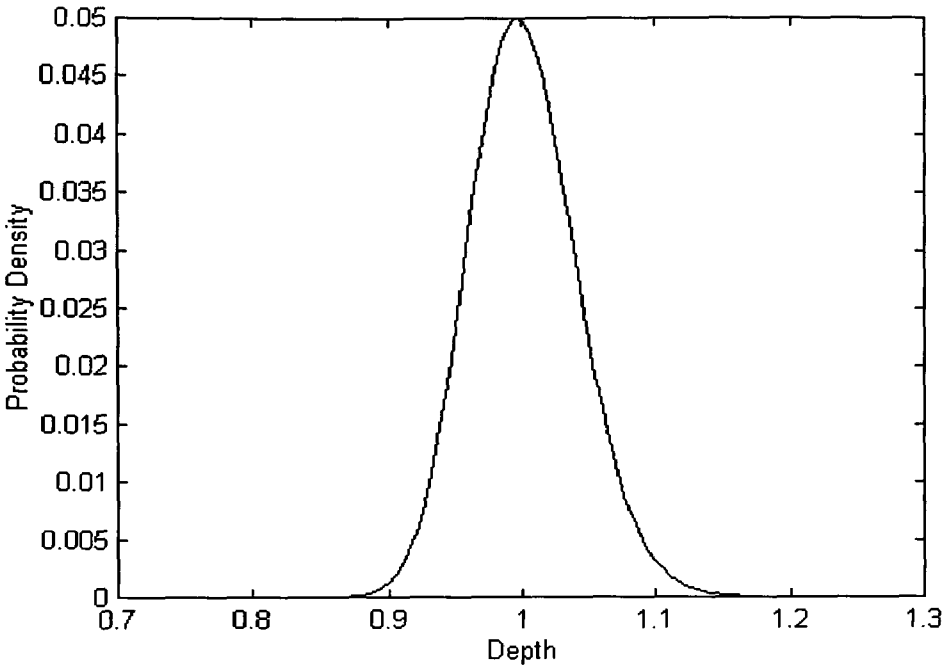

Figure 3.22: Probability distribution of depths corresponding to a Gaussian distribution of disparities for a camera with f = 25mm, p = 4 x 10^{-5} m, a camera translation Δt_x = 4 mm, and a mean disparity of 2.5 pixels with standard deviation 0.1 pixels

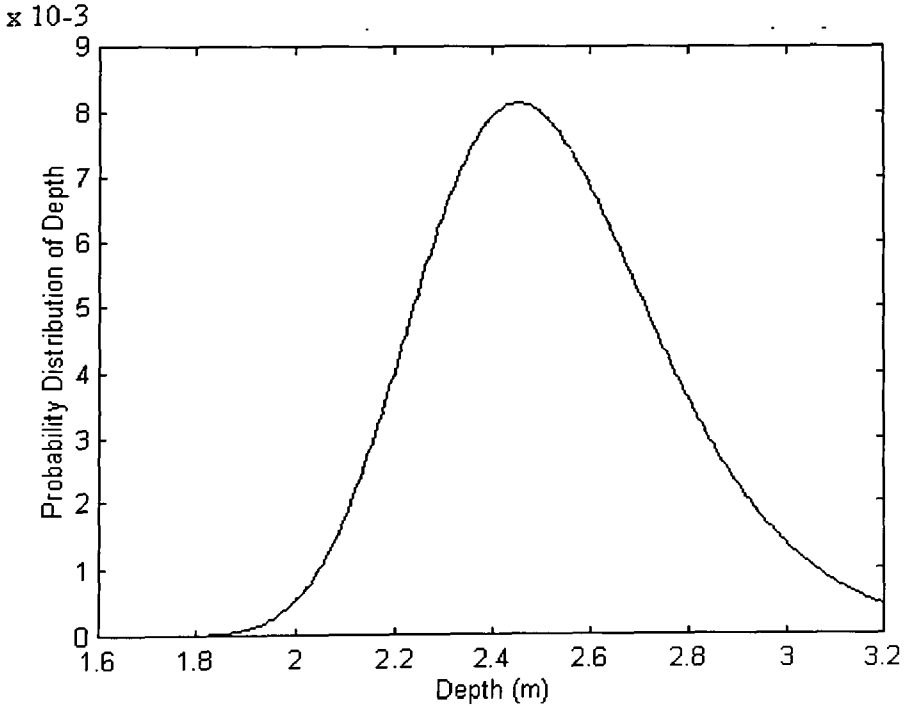

Figure 3.23: Probability distribution of depths corresponding to a Gaussian distribution of disparities for a camera with f = 25mm, p = 4 x 10^{-5} m, a camera translation Δt_x = 4 mm, and a mean disparity of 1.0 pixels with standard deviation 0.1 pixels

3.4 Impulsive noise component

The following section demonstrates the existence and character of an impulsive noise component that corrupts disparity maps generated by the SSD matcher. Figures 3.24 to 3.29 show the distribution of errors for the auto SSD of a 64 x 64 random dot image with uniform distribution of grey scale values, but with progressively more additive white Gaussian noise added to the second image in the pair before matching.

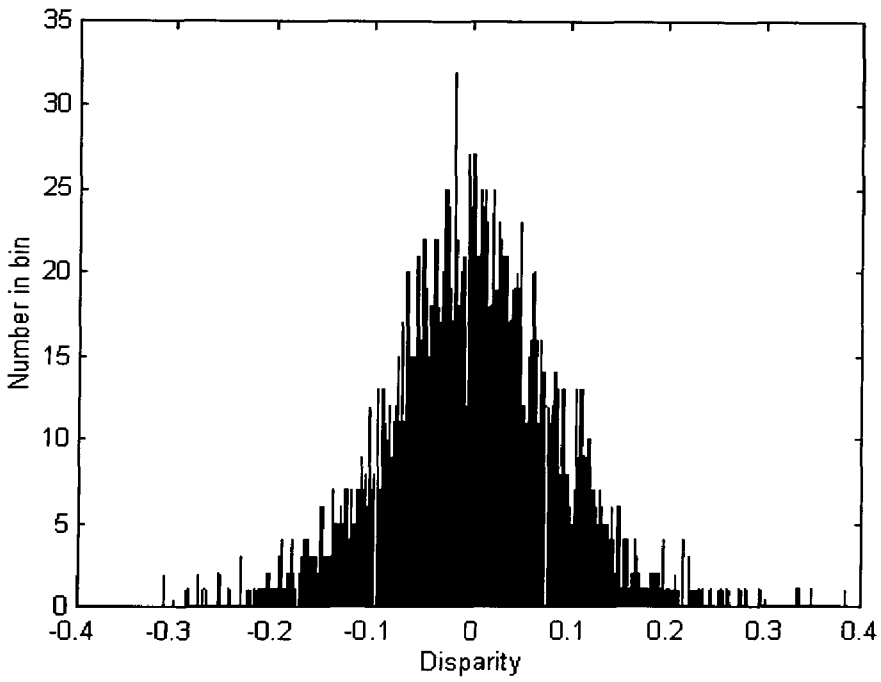

Figure 3.24: Histogram of disparities resulting from a grey-scale image matched with itself (auto SSD) after Gaussian noise of variance =1.0 is added to the grey-scale image

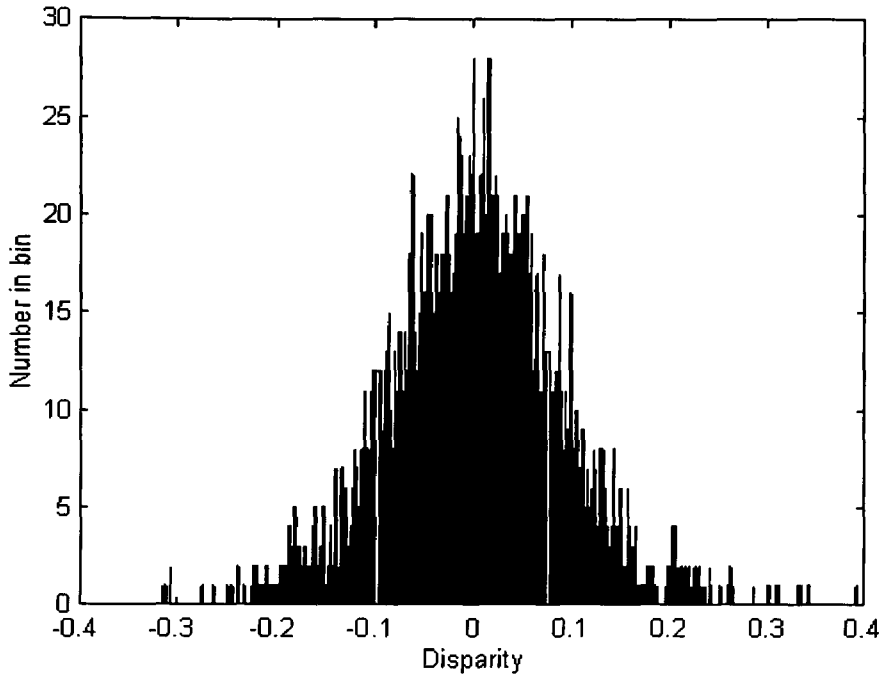

Figure 3.25: Histogram of disparities resulting from a grey-scale image matched with itself (auto SSD) after Gaussian noise of variance =2.0 is added to the grey-scale image

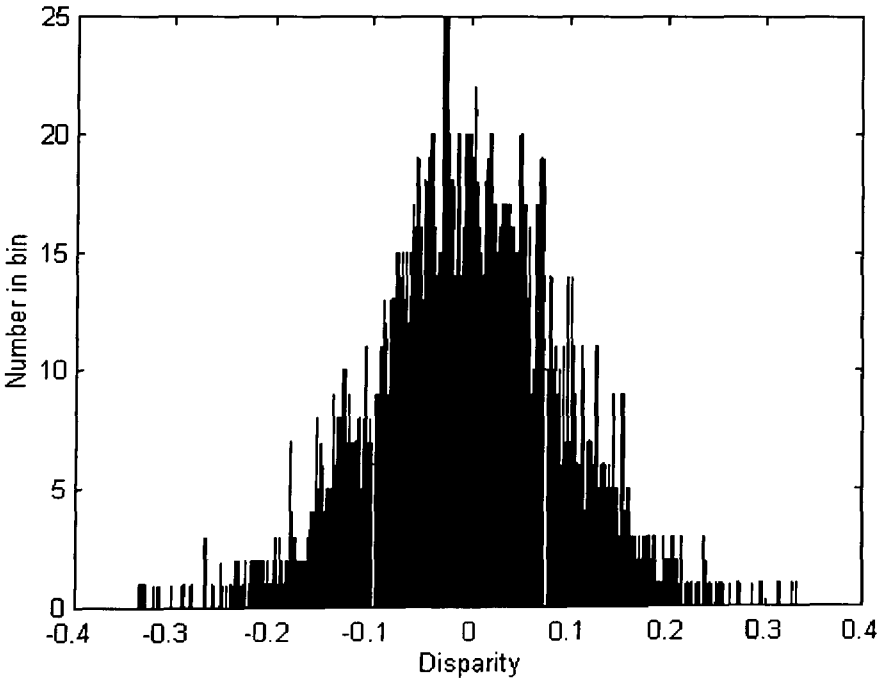

Figure 3.26: Histogram of disparities resulting from a grey-scale image matched with itself (auto SSD) after Gaussian noise of variance =4.0 is added to the grey-scale image

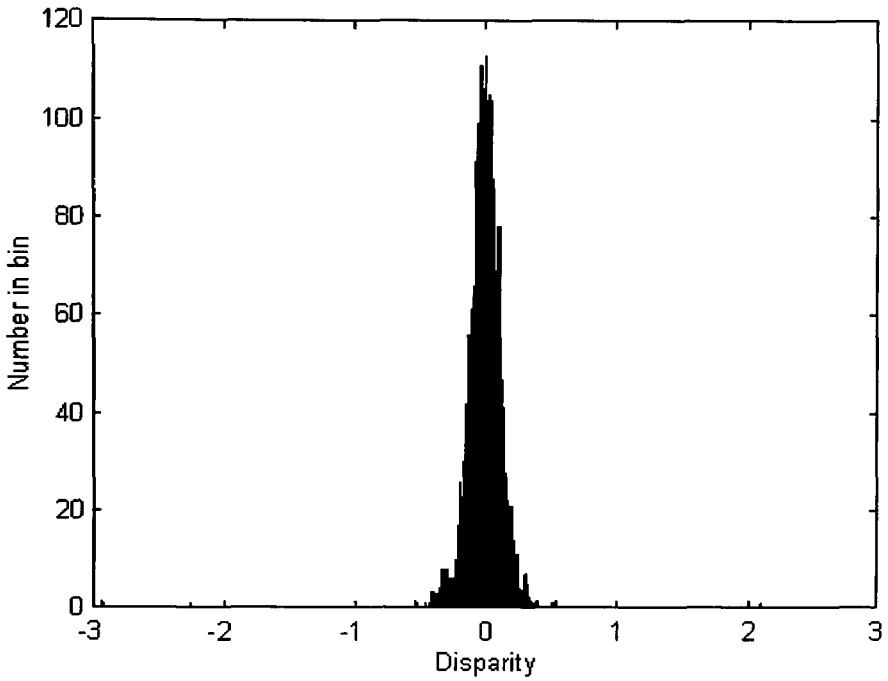

Figure 3.27: Histogram of disparities resulting from a grey-scale image matched with itself (auto SSD) after Gaussian noise of variance =8.0 is added to the grey-scale image

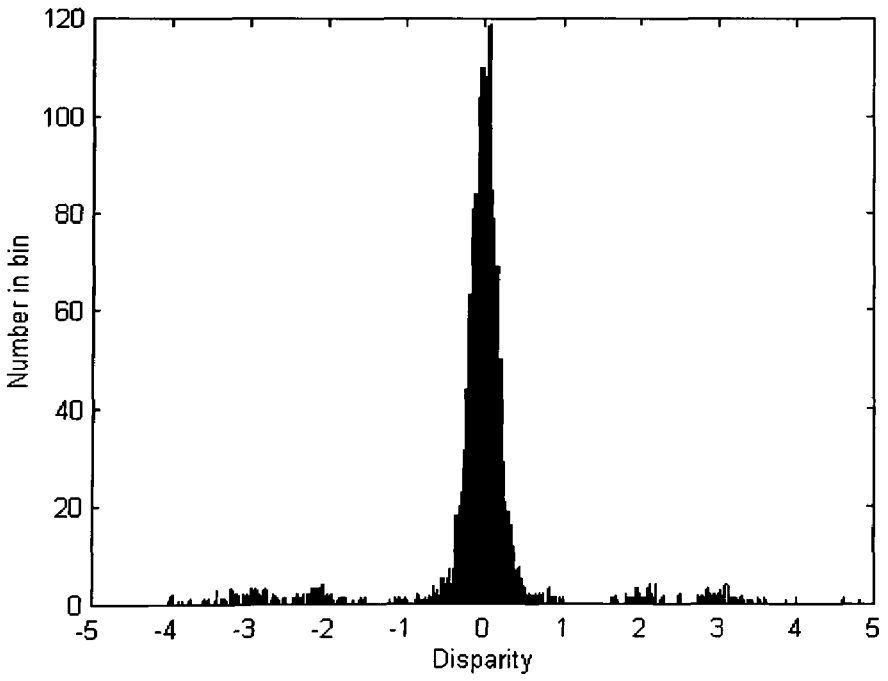

Figure 3.28: Histogram of disparities resulting from a grey-scale image matched with itself (auto SSD) after Gaussian noise of variance = 16 is added to the grey-scale image.

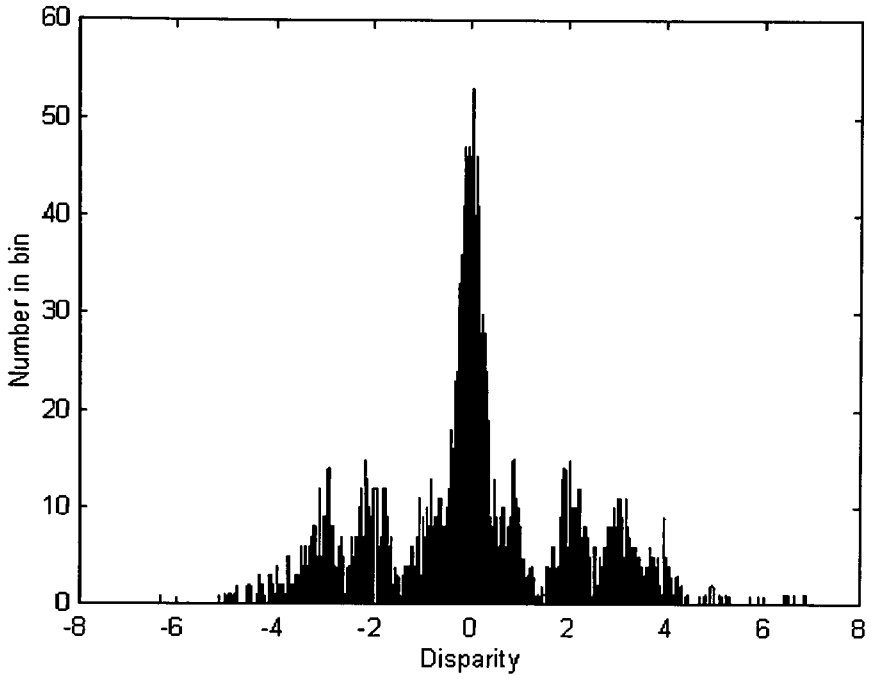

Figure 3.29: Histogram of disparities resulting from a grey-scale image matched with itself (auto SSD) after Gaussian noise of variance = 32 is added to the grey-scale image

It can be seen that initially the added image noise has very little effect on the distribution of errors. For low additive image noise the noise on the disparity is dominated by the backstop noise of the SSD sub-pixel matcher. As more image noise is added however (figure 3.27) then the distribution of errors starts to contain outliers. The position of the minimum SSD shifts from the correct zero offset position to ± 1 pixels, ± 2 pixels, etc. The quadratic is then fitted to these grossly errored positions. The outliers take the form of clusters of errors about integer pixel values. The distribution of the clusters on their own are Gaussian about each integer error for the same reason as the main lobe of the error histogram is Gaussian, but the errors are centred on an incorrect mean value. The errors appear as impulsive noise in the disparity and hence in the depth map produced by the SSD matcher. Figure 3.30 shows the effect of this impulsive noise on a disparity map. The true disparity map for figure 3.30 is a smooth plane, but the plane is corrupted after matching both

by Gaussian noise, making it rough, and by impulsive noise, which produces the spikes on the plane.

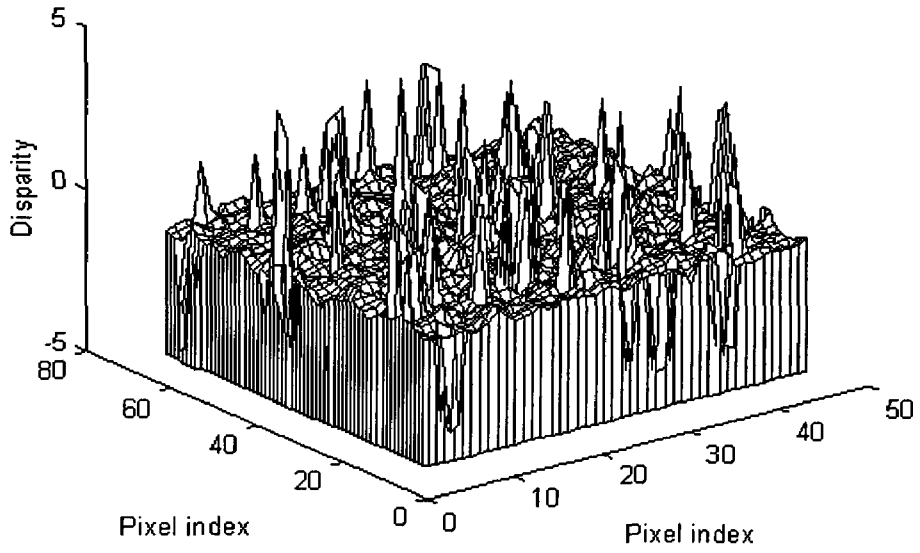

Figure 3.30: Disparity map corrupted by impulsive noise as a result of mismatching due to Gaussian noise of variance = 16 in the grey-scale images from which the depth map is derived

Figures 3.31 to 3.38 show examples of the effect that adding noise to the second image of a pair of grey-scale images that are being matched on the position of the fitted quadratic. The first plot of each pair of plots is the SSD plot and fitted quadratic with no noise added to the second image being matched. The second plot shows the SSD plot and fitted quadratic when noise is added to the second image. Figures 3.31 and 3.32 show a comparatively small effect of the added noise on the disparity. Figures 3.33 and 3.34 show a large effect, resulting in a mismatch and noise spikes. The other figures up to figure 3.38 show similar effects. The estimated error using the approach of (*op cit.* Matthies *et al..*, 1989) is also shown on the plots, showing that this error estimate is unreliable (see also section 3.5).

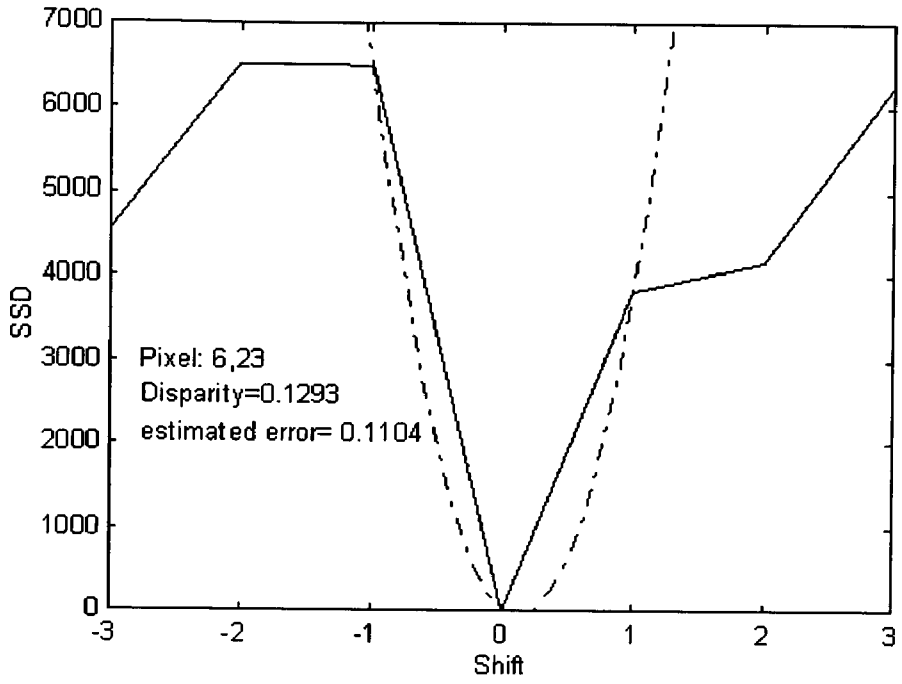

Figure 3.31: Plot showing SSD versus shift and the best fit quadratic for a pair of grey-scale images being matched with no noise added to the grey-scale images

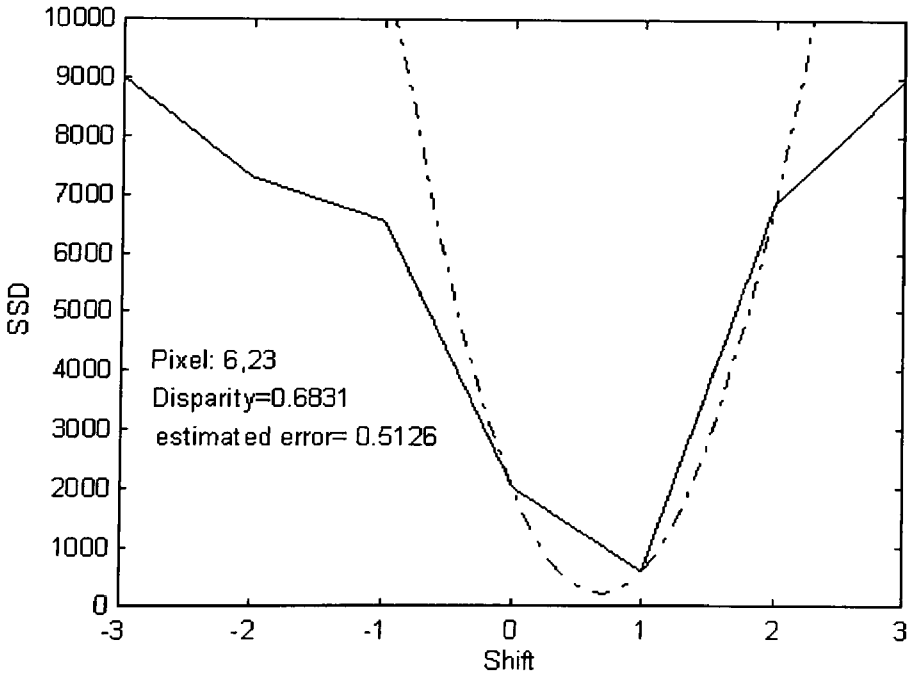

Figure 3.32: Plot showing SSD versus shift and the best fit quadratic for a pair of grey-scale images being matched with noise variance=16 added to the grey-scale images

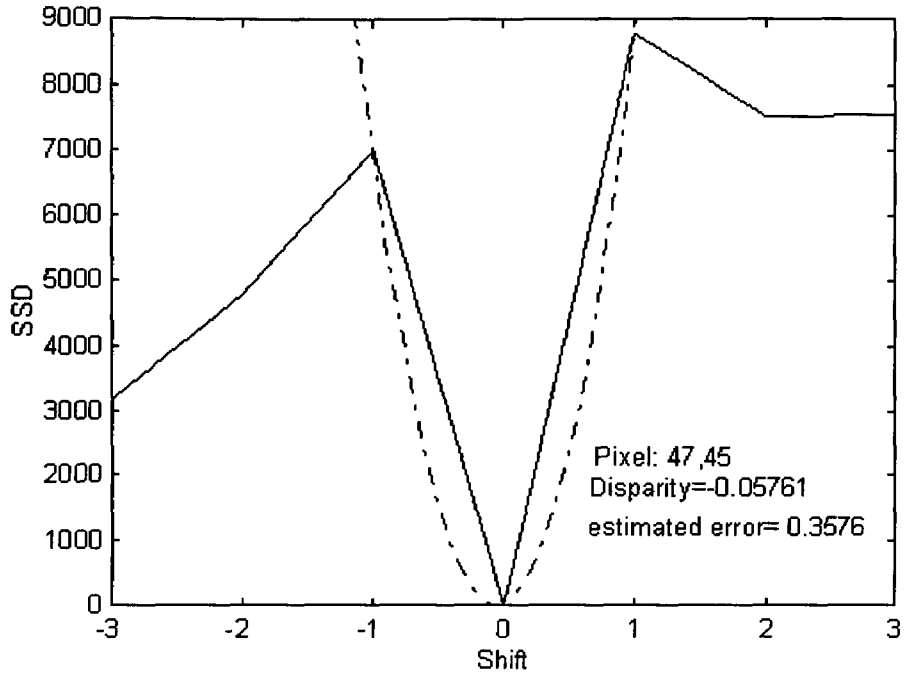

Figure 3.33: Plot showing SSD versus shift and the best fit quadratic for a pair of grey-scale images being matched with no noise added to the grey-scale images

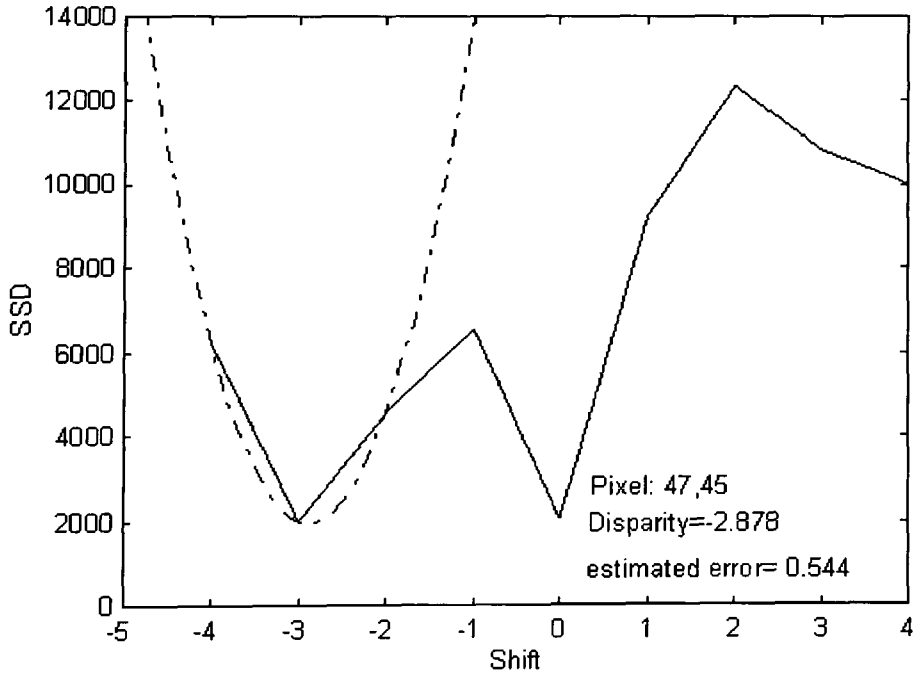

Figure 3.34: SSD versus shift and the best-fit quadratic with noise added to the grey-scale images before matching showing a mismatch, which results in a noise spike

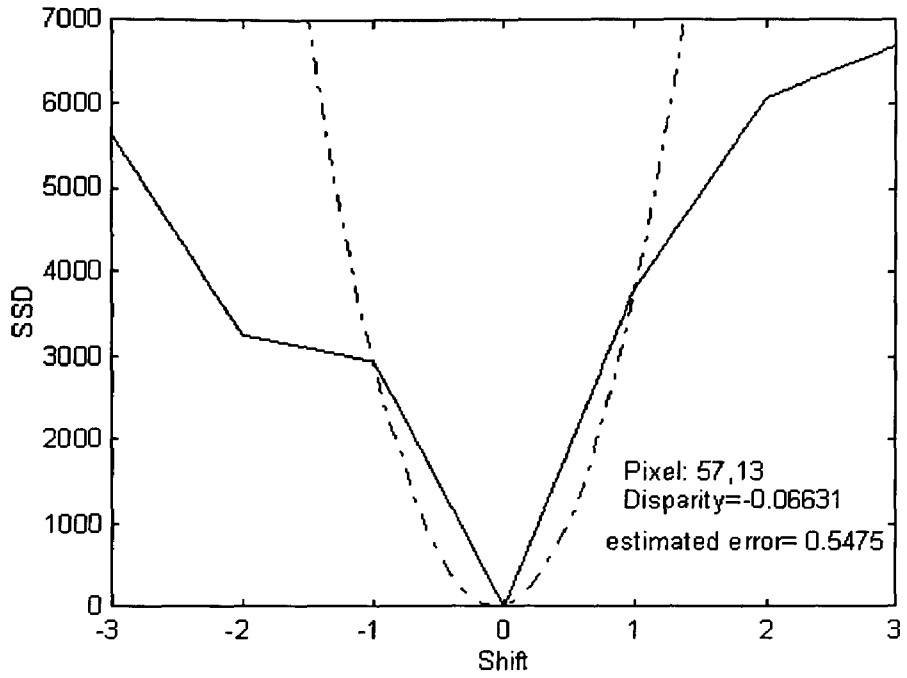

Figure 3.35: Plot showing SSD versus shift and the best fit quadratic for a pair of grey-scale images being matched with no noise added to the grey-scale images

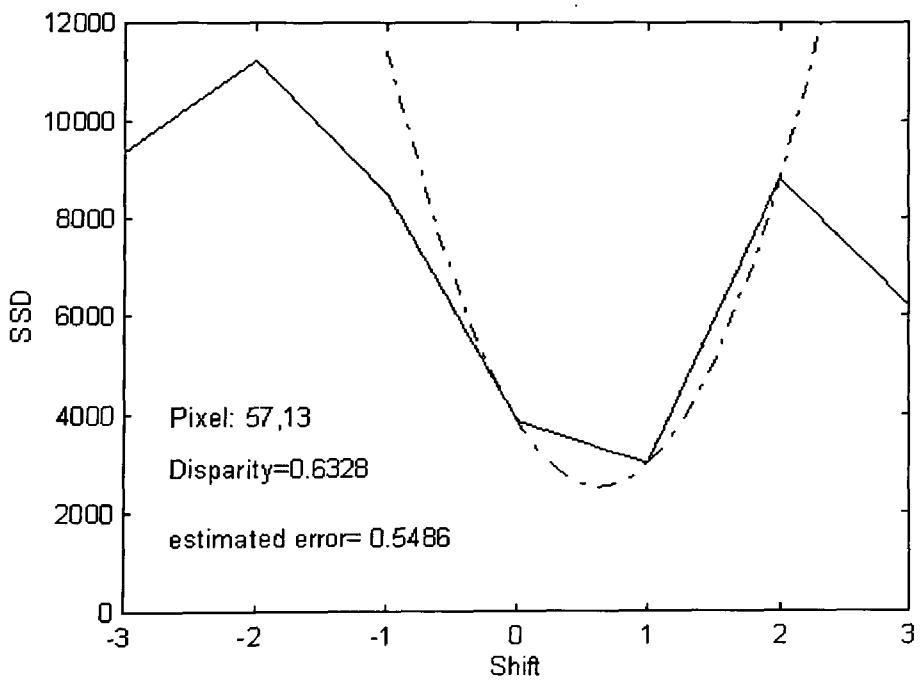

Figure 3.36: SSD versus shift and the best fit quadratic with noise added to the same grey-scale images as used to produce figure 3.35

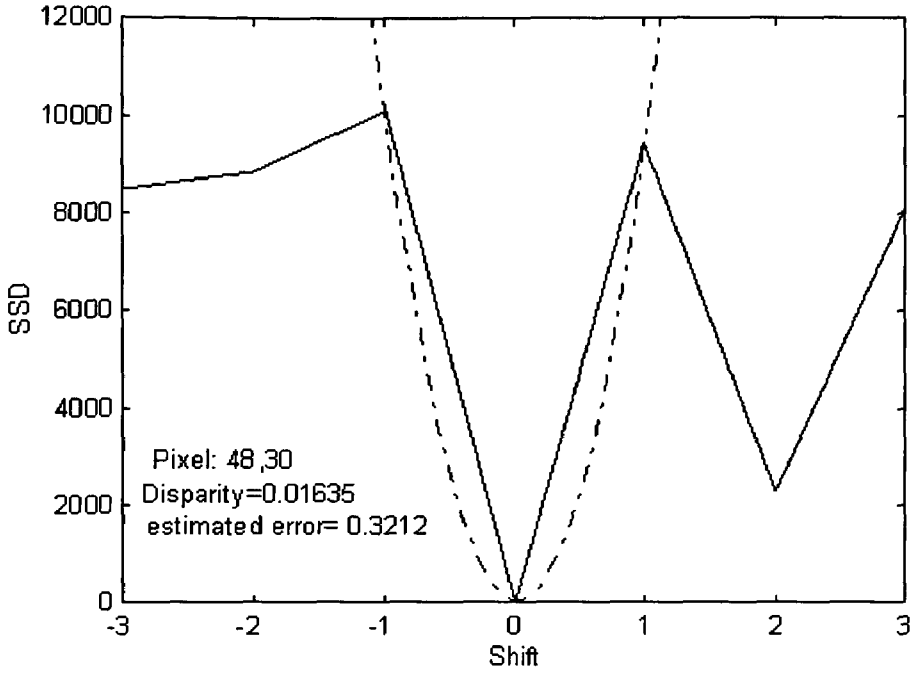

Figure 3.37: Plot showing SSD versus shift and the best fit quadratic for a pair of grey-scale images being matched with no noise added to the grey-scale images.

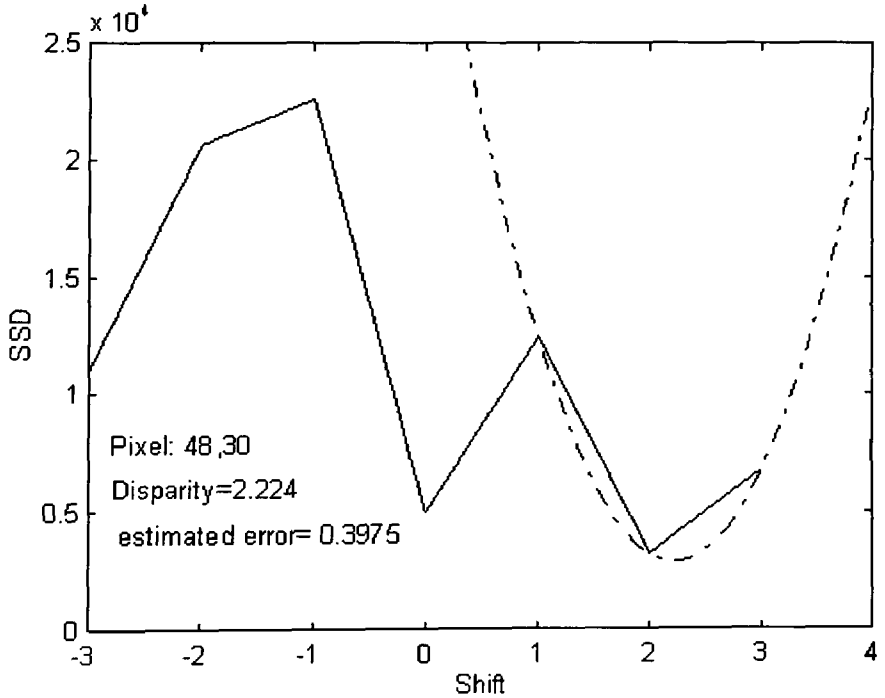

Figure 3.38: SSD plot corresponding to figure 3.37 with added noise, resulting in a mismatch

Figure 3.39 plots the percentage occurrence of impulsive noise in a disparity map versus signal to noise ratio in the grey-scale images used to generate the disparity map. The true disparity is zero pixels and an impulse is deemed to have occurred when the quadratic is fitted around any non-zero disparity value.

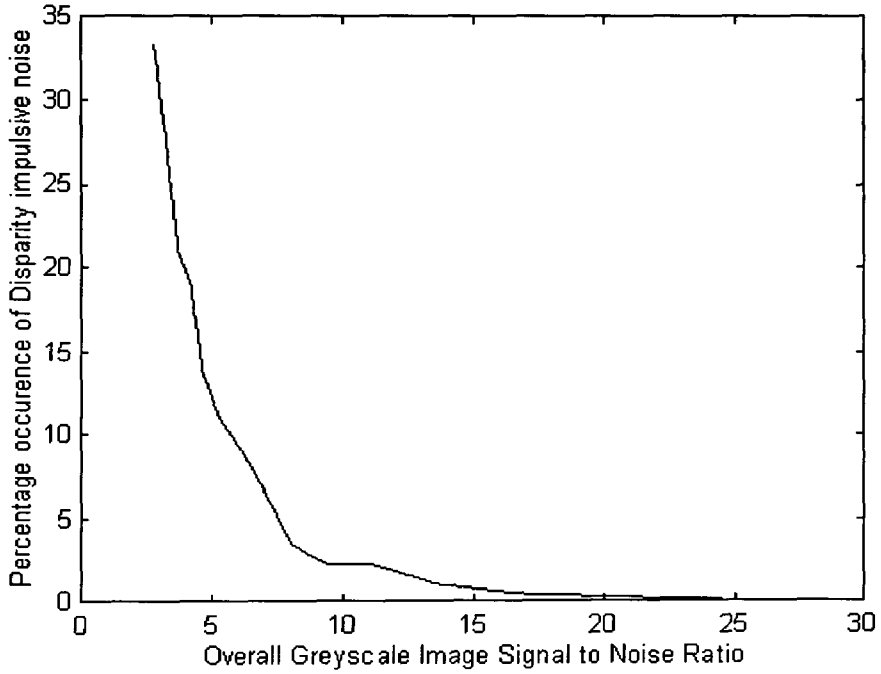

Figure 3.39: Percentage occurrence of impulsive noise in a disparity map versus signal to noise ratio in the grey-scale images used to generate the disparity map

In practice, noise added to the images is only one source of distortion that weakens the implicit constant brightness assumption of correlation based matching. Other distorting effects are changes in illumination, unmodelled lens distortions, perspective distortion, occlusion, and de-occlusion. The effect of noise illustrated above gives a qualitative description of these other distortion effects on the SSD matcher output.

3.5 Uncertainty estimate

Equation 3.14 can be used as the basis for an estimate on the uncertainty in a disparity measurement, and hence the uncertainty in the corresponding depth map. This uncertainty measure is needed by the Kalman filter described in section 2.3.4. It is not clear from (Matthies *et al.*, 1989) whether that work used a fixed value for σ_n^2 in equation 3.14 It is also stated there that the coefficient $c(u, v)$ is a chi-squared distributed random variable with mean σ_n^2. It is suspected that the number of degrees of freedom of this chi-squared distribution is too low for the mean to be used as a means for evaluating σ_n^2. In practice, in some cases the coefficient of the squared term in the SSD plot, $a(u, v)$ is large, but the residual SSD at the minimum is also large. This gives a low estimate of uncertainty, which is fed to the Kalman filter, when a large residual SSD estimate at the minimum argues that the estimate of disparity is poor. An estimate of uncertainty based on three heuristic measures has been proposed (*op cit* Corbatto *et al.*, 1995), and in more detail (*op cit* Trucco *et al.*,1996). These measures are the absolute SSD error, a measure of irregularity of the SSD plot about the minimum, and a measure of the peak to peak local variation. The three measures are combined to give a quasi variance estimate. In the work described in this thesis, however, the uncertainty measure of equation 3.14 using a fixed σ_n^2 is fed to the Kalman filter where it is used since there is some more theoretical basis to this uncertainty measure than the more heuristic approach.

3.6 Ill-posedness, regularisation, and filtering of disparity maps

3.6.1 Ill-posedness and stereo matching

Sections 3.3 and 3.4 examine the SSD matcher from an additive noise viewpoint. However the behaviour of the SSD matcher can be viewed from a different perspective. Many inverse problems such as the recovery of depth information using stereo matching fall into the mathematical class of ill-posed problems (Poggio *et al.*1985), (Terzopoulos 1986(b)), (Bertero *et al.*1988). Ill-posed problems are those which satisfy at least one of the following conditions:

- A solution does not exist (non-existence)
- More than one possible solution exists (non-uniqueness)
- The solution does not depend continuously on the data (non-continuity)

In general the SSD matcher can satisfy any of the above conditions as illustrated in the following examples:

- In an area of no texture the matcher cannot find a minimum SSD value (non-existence)
- In an area of a repeating pattern there will be more than one equal minimum (non-uniqueness)
- Where noise is added to the grey-scale image pair being matched or where distortion due to perspective occurs the output of the SSD matcher changes in a discontinuous way.

It is the occurrence of the last of these conditions that leads to the impulsive noise that has been observed and discussed in section 3.4. It may be that impulsive noise is a hallmark for processes involving ill-posed problems for which the solutions are unstable to noise.

3.6.2 The role of prior assumptions in regularisation

The classical way of dealing with ill-posed problems is to impose additional constraints to the problem formulation such that the best solution is chosen given uncertain and ambiguous

measurements. These additional constraints express *a priori* assumptions about the form of acceptable solutions. These *a priori* constraints can be expressed in variational terms.

In the case of depth and disparity maps the most commonly used constraint is that of local smoothness. In a typical viewed scene the change of disparity per pixel or disparity gradient for the majority of pixels is low. Thus, apart from at a finite number of discontinuities the disparity gradients and hence the depth gradients are limited. (Hakkarainen *et al.*, 1991) and later (Kanade and Okutomi, 1994) review the development of local smoothness constraints from the frontoparallel constraint of (Marr and Poggio, 1976), to the disparity gradient limits of (Grimson, 1985) and (Pollard *et al.*, 1985). (*op cit* Kanade and Okutomi, 1994)use a constraint that includes a measure of the proximity of pixels as well as the disparity gradient between them.

3.6.3 Equivalence of regularisation to filtering

The application of regularising prior constraints to ill-posed problems can be regarded as equivalent to a filtering process when those constraints are expressed in variational terms. That this is so is demonstrated by the following argument of (*op cit.* Terzopoulos 1986(b)). Consider a one-dimensional signal, $c(x)$ which is the noise-corrupted output signal of an unconstrained process (e.g. the raw output of an SSD sub-pixel matcher). A simple smoothness constraint would be to look for a solution $s(x)$ which over the domain of the signal (all x) minimised the integral of the weighted sum of the square of the first derivative of the solution and the squared difference of the solution and original noisy signal. This results in the following functional to be minimised:

$$E = \int \left(s(x) - c(x)\right)^2 + \lambda \left(\frac{d}{dx}[s(x)]\right)^2 dx \qquad 3.38$$

The Euler-Lagrange equation whose solution for $s(x)$ minimises E in 3.38 is:

$$s(x) - c(x) - \lambda \frac{d^2}{dx^2}[s(x)] = 0 \qquad 3.39$$

Taking the Fourier Transform of 3.39 gives:

$$S(\omega) - C(\omega) - \lambda(j\omega)^2 S(\omega) = 0 \qquad 3.40$$

Which is the low pass filter form:

$$S(\omega) = \frac{C(\omega)}{1 - \lambda(j\omega^2)} \qquad 3.41$$

So that the process of minimising the functional 3.38 is equivalent to filtering the corrupted data with a low pass filter. As is pointed out by (*op cit.* Terzopoulos 1986(b), constraints that modify their behaviour depending on the form of the signal do not have a linear form and a frequency domain representation like equation 3.41. However, there is a general correspondence between filtering and the regularisation of an ill-posed problem, although the filters may be complex and non-linear.

Most types of filtering or signal estimation processes involve the implicit application of general constraints that are based on models of the true underlying signal and noise process. Where the model is tightly applied then high efficiency or even optimality is obtained for signals and noise processes that fit the model. Where the assumptions of the model are violated, however, the performance can degrade dramatically. Looser signal models are more robust. The question that this thesis explores is whether a filter framework based on fuzzy logic can effectively apply broad regularisation constraints in an explicit and robust manner. The problem of depth map filtering,

viewed as a regularisation problem, requires a class of filters that can capture complex and imprecise regularisation constraints. The ability of fuzzy systems to capture complexity and uncertainty motivates the question of their utility as a basis for such a class of filters.

3.7 Summary of Chapter 3.

The purpose of this chapter has been to examine the noise processes that arise from the generation of disparity and depth maps using the Sum Squared Differences matcher technique. The Sub-pixel matcher of (*op cit.* Anandan, 1984) and (*op cit.* Matthies *et al.*, 1989) has been described, and the noise processes associated with this matcher have been examined. It is shown that the noise processes are a mixture of (almost) Gaussian noise and impulsive noise. The existence of a 'backstop' Gaussian noise for this type of matcher, not discussed in either (*op cit.* Anandan, 1984) (*op cit.* Matthies *et al.*, 1989) and demonstrated but not analysed in (*op cit.* Trucco *et al.*,1996), is established and explained. This backstop noise is in addition to any Gaussian noise induced by noise in the original grey-scale images and is associated with the quadratic-fitting sub-pixel matcher. The chapter has also reviewed the errors associated with stereo matching from the perspective established by (*op cit.* Poggio *et al.*1985) of the ill-posed nature of stereo matching by correlation. The equivalence in principle between filtering of signals and the application of regularisation to ill-posed problems is also briefly reviewed. This equivalence is a major motivation for adopting the approach taken in this thesis of using filters based on fuzzy logic. In such filters the stabilising prior assumptions applied to regularise the ill-posed problem can be explicitly encoded in the fuzzy inferencing system. The next chapter, Chapter 4, reviews other major approaches which can be used to filter depth or disparity maps corrupted by mixed noise.

3.8 References.

(Anandan, 1984) Anandan P. "Computing dense displacement fields with confidence measures in scenes containing occlusion. " Proceedings DARPA image understanding workshop. pp 236-246, 1984.

(Anandan, 1989) Anandan, P. "A Computational Framework and an Algorithm for the measurement of visual motion." International Journal of Computer Vision. Vol 2 pp. 283-310. 1989.

(Bertero *et al.*1988) Bertero, M. Poggio, T. and Torre V. "Ill Posed Problems in Early Vision." Proceedings of the IEEE Vol. 76 No. 8 pp 869 - 889, Aug. 1988.

(Corbatto *et al.*1995) Corbatto M. Tinonin S. Trucco E. and Roberto V. "Dense Depth Maps From Motion Using Dynamic Data Fusion". Proceedings of IEE Conference on Image Processing and its Applications. pp. 642-646. July 1995.

(Davis, 1963) Davis P.J. "Interpolation and Approximation." Blaisdell 1963.

(Grimson, 1985) Grimson W. "Computational Experiments with a Feature Based Stereo Algorithm." IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 7 No. 1 pp 17-34 Jan 1985.

(Hakkarainen *et al.*, 1991) Hakkarainen J. M., Little J.J., Lee H-S., and Wyatt J.L. "Interaction of Algorithm and Implementation for Analog and VLSI Stereo Vision." SPIE Visual Information Processing: From Neurons to Chips Vol. 1473 pp 173-184 1991.

(Kanade and Okutomi, 1994) Kanade T., and Okutomi M. "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment." IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 16 No 9 September 1994.

(Kendall and Stuart 1963) Kendall M.G. and Stuart A. "The Advanced Theory of Statistics" vol. 1 2nd Ed. (Three-volume edition). Griffin. 1963.

(Marr and Poggio, 1976) Marr D. and Poggio T. "Cooperative computation of Stereo disparity" Science Vol. 194 pp 283-287. Oct 1976.

(Matthies *et al.*, 1989) Matthies L., Kanade T.,and Szeliski R., "Kalman Filter-based Algorithms for Estimating Depth from Image Sequences." International Journal of Computer Vision. No. 3, pp 209-236, 1989.

(Poggio *et al.*1985) Poggio T. Torre V. Koch C. "Computational vision and regularisation theory." Nature Vol. 317. Sept 1985.

(Pollard *et al.*, 1985) Pollard S., Mayhew J., and Frisby J. "PMF A stereo correspondence algorithm using a disparity gradient limit." Perception Vol 14 pp 449-470, 1985.

(Terzopoulos 1986(b)) Terzopoulos D. "Regularisation of Inverse Visual Problems Involving Discontinuities." IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 8 No. 4 pp 413 - 424, July 1986.

(Trucco *et al.*,1996) Trucco E, Roberto V, Tinonin S and Corbatto M. "SSD Disparity Estimation for Dynamic Stereo", Proceedings British Machine Vision Conference Vol. 1, pp 343-352. 1996.

Chapter 4 : Major Current Approaches to Filtering and Smoothing

4.1 Introduction

The noise processes associated with the SSD matcher have been discussed and highlighted in Chapter 3. These noise processes result in a mixed Gaussian and impulsive noise corrupting the disparity map, and this mixed noise is further transformed on conversion of the disparity map to a depth map. The impulsive component of the noise is associated to the fact that stereo matching is an ill-posed problem, which requires regularisation in order to obtain stable solutions. The correspondence between regularisation and filtering was discussed in section 3.6.3. This chapter is a review of existing approaches that could be taken to filter either the two-dimensional disparity or the two-dimensional depth map. The review serves three purposes.

One purpose is to establish the relationship of the fuzzy filter approach adopted in this thesis to existing methods. In particular the unifying concept is suggested that filtering processes correspond to the mathematical idea of a function. Since it has been shown (Wang, 1992), (Kosko, 1992), and (Kosko, 1994) that fuzzy inferencing systems can be universal function approximators, then fuzzy inferencing systems should in principle be able to approximate any filtering process that can be described as a function. Another link between the fuzzy filter approach and existing approaches is that fuzzy filters provide an alternative description of prior assumptions about the underlying image being filtered to the prior statistical distribution used by regularisation-based techniques.

The second reason for examining existing filter methods is that the fuzzy logic based approach to filtering taken in this thesis allows the possibility of the integration of two or more of these existing filtering techniques within a unified filter architecture.

The third purpose for reviewing different approaches to filtering is that three existing techniques, the Wiener, moving average, and median filters are used as benchmarks with which to compare the performance of the novel filters described in this thesis

4.2 Black box model of a digital filter.

In the filters described in this thesis, all inputs or outputs are regarded as real numbers. It is assumed that these real numbers are represented within computations as floating point numbers of an adequate precision to the task in hand. The real numbers are samples drawn from a 'signal' which may be one or two-dimensional and which represents some real world property such as depth in a scene. Within the thesis the term 'filter' is viewed in a broad sense as a 'Black Box' process which maps an input vector of real numbers into a single output value. The single output value is used to replace one of the input vector values. The input vector is arranged to consist of samples of the signal drawn in some (usually symmetrical) way from the neighbourhood surrounding the input value to be replaced. The neighbourhood is referred-to as the 'filter window'. Figure 4.1 shows how an input vector can be extracted from a two-dimensional image signal for presentation to a filter using a square 3 x 3 filter window. Other possible filter window shapes are rectangular and cross-shaped.

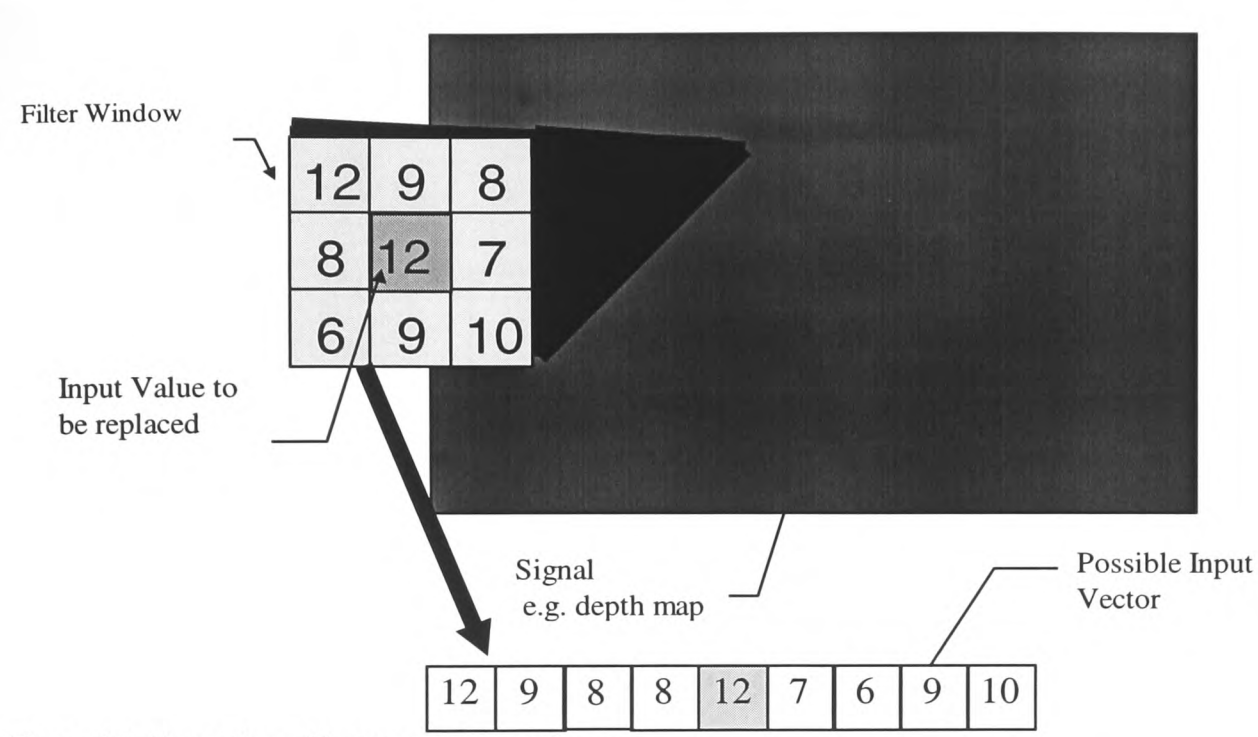

Figure 4.1: Extraction of input vector from image

Figure 4.2 illustrates the notion of a filter as a Black box. For a filter, the mapping performed by the Black box is a function in N dimensions from a domain of real numbers onto a one-dimensional co-domain as the correspondence from domain to co-domain has to be unique. Moreover for the type of filters considered in this thesis the domain and co-domain are the same set (generally positive real numbers), therefore the Black box satisfies not only the mathematical definition of function (Brainerd *et al.*, 1967) but also that of an operation.

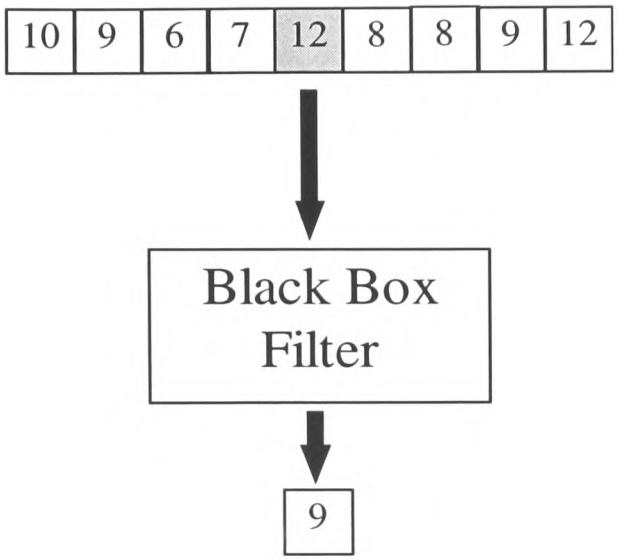

Figure 4.2 Black Box View of filter

In summary, a general filter is essentially a function. For the problem domain addressed in this thesis the function also needs to be an operation. Since the idea of a function is intimately tied to that of sets, it is observed that a generalised view of filters can be formulated in terms of set theory.

4.3 Linear Filters

4.3.1 Introduction

Linear filters are the most commonly used class of digital filter. Linearity implies that the function performed by these filters is independent of the amplitude of the input signal. Strictly linear filters are temporally or spatially invariant. Those filters that vary with time or space are termed adaptive. Since in general adaptive filters alter their behaviour in response to changes in the incoming signal such filters are often not strictly linear, but may be locally linear.

4.3.2 FIR and IIR low pass filters

A class of non-recursive filters that is frequently used for signal smoothing is the Finite Impulse Response (FIR) class of filters. The diagram for a discrete FIR filter operating in one dimension is shown in figure 4.3:

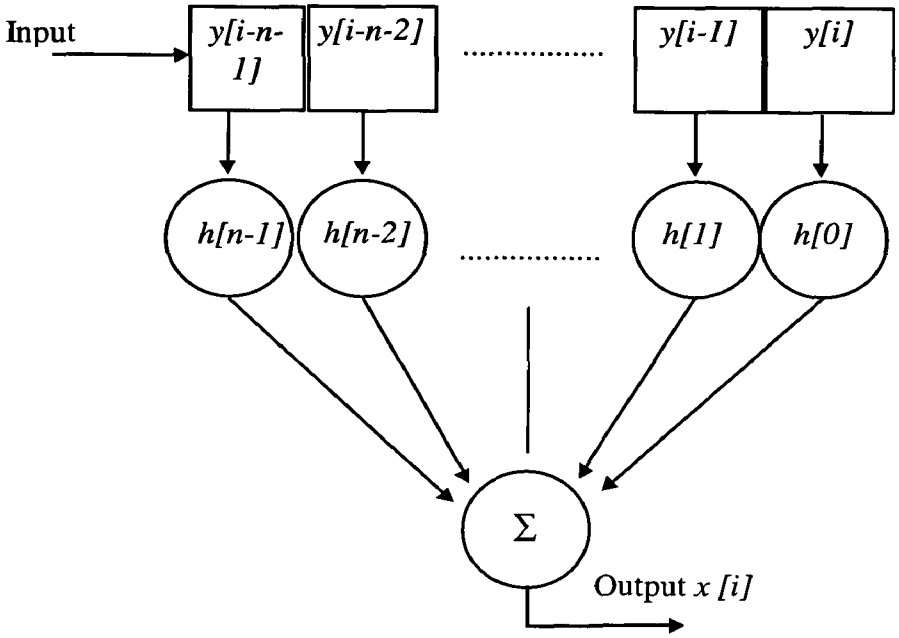

Figure 4.3: General one-dimensional FIR filter

The FIR filter generates an output $\hat{x}[i]$ which is the weighted sum of the present and past inputs, $y[i]$ to $y[i-n]$. The vector of weights, $h[i]$ is the impulse response of the filter. The output is given by:

$$\hat{x}[i] = \sum_{k=0}^{n-1} h[k]y[i-k] \qquad 4.1$$

In the two-dimensional case, where each input pixel value in an n x m rectangular filter window is represented by $y(k,l)$, each output value is $\hat{x}[i,j]$, and the impulse response is replaced by the Point Spread Function, $h(k,l)$ the equation 4.1 becomes:

$$\hat{x}[i,j] = \sum_{k=-\frac{(n-1)}{2}}^{\frac{n-1}{2}} \sum_{l=-\frac{(m-1)}{2}}^{\frac{m-1}{2}} h[k,l]y[i-k,j-l] \qquad 4.2$$

In this case the Finite Point Spread (FPS) filter replaces the centre pixel value by a weighted sum of its neighbours. For the case where all the $h[k,l]$ are equal to $\frac{1}{nm}$ the filter becomes a moving average filter. Both the FIR and FPS filter are linear filters and have a corresponding one and two-dimensional frequency domain representation respectively. A linear filter is effective in removing noise if it attenuates strongly at those frequencies where the noise has a high level and the signal a low level. At frequencies where the converse is the case the filter should have low attenuation. Such linear filters operate on the *a priori* assumption that the signal and the noise can be distinguished in the frequency domain. Where the signal and noise frequency components overlap in the frequency domain the filter must make a compromise. A type of linear filter that uses *a priori* knowledge of the signal and noise statistics to optimise this compromise is the Wiener filter (Wiener, 1949).

FIR filters are non-recursive filters. If the filter output is a linear function of both the past inputs and outputs of the filter the filter becomes an infinite impulse response (IIR) filter. Such filters are computationally more efficient. However it is also more difficult to ensure stability of the filter, particularly for two-dimensional signals. Since the fuzzy filters presented in this thesis are based

on a non-recursive type of architecture IIR filters are not discussed further, although a recursive architecture could be used within the framework of the fuzzy filters described in Chapter 6.

4.3.3 Optimal Linear filters - Wiener Filters

The Wiener filter is an optimum non-recursive linear estimator (*op cit* Wiener 1949). (FIR/FPS filter) for a Gaussian signal corrupted by Gaussian noise. The criterion for optimality is minimum mean squared error. In the following description the true pixel values at position indices *i,j* are represented by *x[i,j]*, the noise corrupted pixel values by *y[i,j]* and the estimate produced by the filter as $\hat{x}[i,j]$. *E[.]* is the expected value operator. The mean squared error, ε, to be minimised is given by:

$$\varepsilon = E\{(x[i,j] - \hat{x}[i,j])^2\} \qquad 4.3$$

The estimate $\hat{x}[i,j]$ is given by equation 4.2. The minimum mean squared error will occur when all the partial derivatives of the mean squared error with respect to the point spread function coefficients are set to zero so that:

$$\frac{\partial \varepsilon}{\partial h(p,q)} = 2E\{(x[i,j] - \sum_{k=-\frac{(n-1)}{2}}^{\frac{n-1}{2}} \sum_{l=-\frac{(m-1)}{2}}^{\frac{m-1}{2}} h[k,l]y[i-k,j-l])y[p,q]\} = 0 \qquad 4.4$$

which can be re-written as an orthogonality constraint for all pixel pairs *[i,j]* and *[p,q]:*

$$E\{(x[i,j]-\hat{x}[i,j])y[p,q]\}=0 \tag{4.5}$$

It can also be written as:

$$E\{x[i,j].y[p,q]\}=E\{\sum_{k=\frac{(n-1)}{2}}^{\frac{n-1}{2}}\sum_{l=\frac{(m-1)}{2}}^{\frac{m-1}{2}}h[k,l]y[i-k,j-l])y[p,q]\} \tag{4.6}$$

If the true image *x[i,j]* and the observed image *y[p,q]* are jointly stationary then the cross correlation depends only on the difference between the image co-ordinates, *i-p,j-q*.

In the case of additive Gaussian noise of mean zero which is uncorrelated with the uncorrupted signal:

$$y[i,j]=x[i,j]+n[i,j] \tag{4.7}$$

and:

$$E\{x[i,j].y[p,q]\}=E\{x[i,j].x[p,q]\}=R_{xx}(i,j) \tag{4.8}$$

Where $R_{xx}(i,j)$ is the autocorrelation of the uncorrupted signal. Also:

$$E\{\sum_{k=\frac{(n-1)}{2}}^{\frac{n-1}{2}}\sum_{l=\frac{(m-1)}{2}}^{\frac{m-1}{2}} h[k,l]y[i-k,j-l].y[p,q]\}$$

$$= E\{\sum_{k=\frac{(n-1)}{2}}^{\frac{n-1}{2}}\sum_{l=\frac{(m-1)}{2}}^{\frac{m-1}{2}} h[k,l](\quad (x[i-k,j-l]+n[i-k,j-l]).(x[p,q]+n[p,q])\quad)\} \qquad 4.9$$

$$= \sum_{k=\frac{(n-1)}{2}}^{\frac{n-1}{2}}\sum_{l=\frac{(m-1)}{2}}^{\frac{m-1}{2}} h[k,l](R_{xx}[i-k,j-l]+R_{nn}[i-k,j-l])$$

By the Wiener-Khinchine relation, the autocorrelation and the power spectral density of wide sense stationary processes are Fourier transform pairs, therefore taking the discrete Fourier transform of equation 4.9 above and rearranging for the two dimensional frequency transfer function of the filter, $H[\omega_k,\omega_l]$:

$$H[\omega_k,\omega_l] = \frac{PSD_x[\omega_k,\omega_l]}{PSD_X[\omega_k,\omega_l]+PSD_n[\omega_k,\omega_l]} = \frac{1}{1+PSD_n[\omega_k,\omega_l]/PSD_x[\omega_k,\omega_l]} \qquad 4.10$$

This is the frequency domain description of the Wiener filter for a signal corrupted by uncorrelated Gaussian noise. In parts of the spectrum where the noise power density is small compared to the signal power density this filter has a gain of unity. In parts of the spectrum where the noise PSD dominates the filter has a small gain. If the noise PSD is small at all frequencies then the filter has a gain of unity at all frequencies and a corresponding point spread function which is a Dirac delta function at the origin. i.e. each pixel value is left unchanged. For the case of a true signal which is a constant flat surface corrupted by zero mean Gaussian noise, the filter frequency domain response is a Dirac delta at the frequency origin and zero elsewhere. Therefore, for this case, the Wiener

filter point spread function becomes an averaging operation over all pixels. This averaging operation can be approximated by a moving average filter. Thus a moving average filter implicitly assumes a locally flat surface depth map model and zero mean Gaussian noise. Locally flat depth map models contain most energy at low spatial frequencies. A common statistical model used for images and depth maps is a Markov process (Geman and Geman, 1984), (Li, 1994), in which the conditional probability of a pixel taking a certain value depends only on the values of its neighbours (Poggio *et al.*.1985). A first order Markov field has a power spectral density given by:

$$PSD_x(\omega_k,\omega_l) = \frac{1}{\sqrt{\alpha_k \alpha_l}} \frac{2K}{1+[\omega_k^2/\alpha_k^2+\omega_l^2/\alpha_l^2]} \qquad 4.11$$

where K ,α_k, and α_l are constants. For small α_k, and α_l this PSD model tends towards the spike at the origin, and the optimal filter tends towards a moving average filter.

Example

Figure 4.4 shows the simulated depth map "cake2" and figure 4.5 shows "cake2" corrupted by additive Gaussian noise. Figure 4.6 shows the resulting two-dimensional Wiener filter transfer function whilst figure 4.7 shows the depth map after Wiener filtering. In the cases shown below the noise and signal power spectra are known exactly *a-priori* which is not the case for real depth maps. In general some estimate of the signal and noise PSD have to be made. If these are incorrectly estimated then the performance of the filter deteriorates badly.

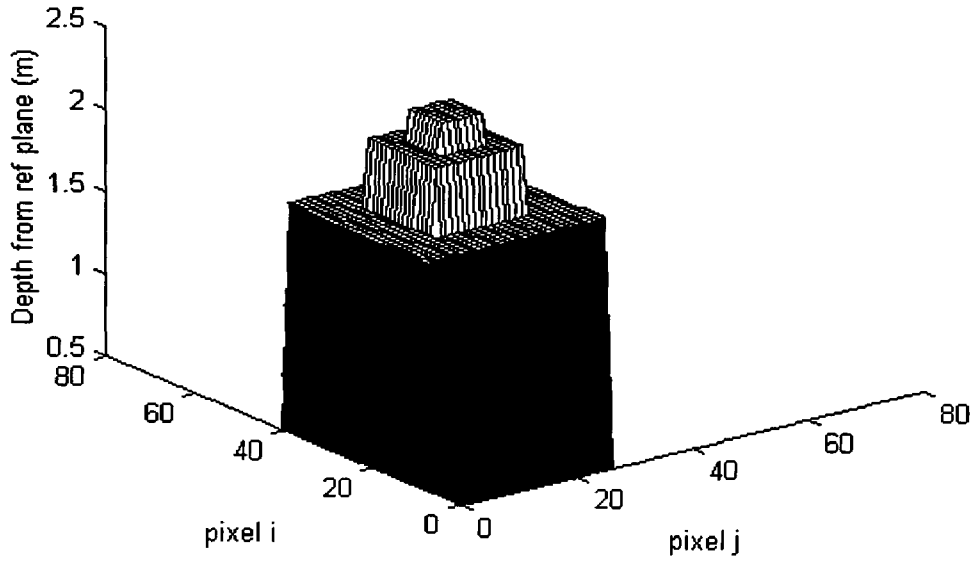

Figure 4.4 Simulated depth map "cake2

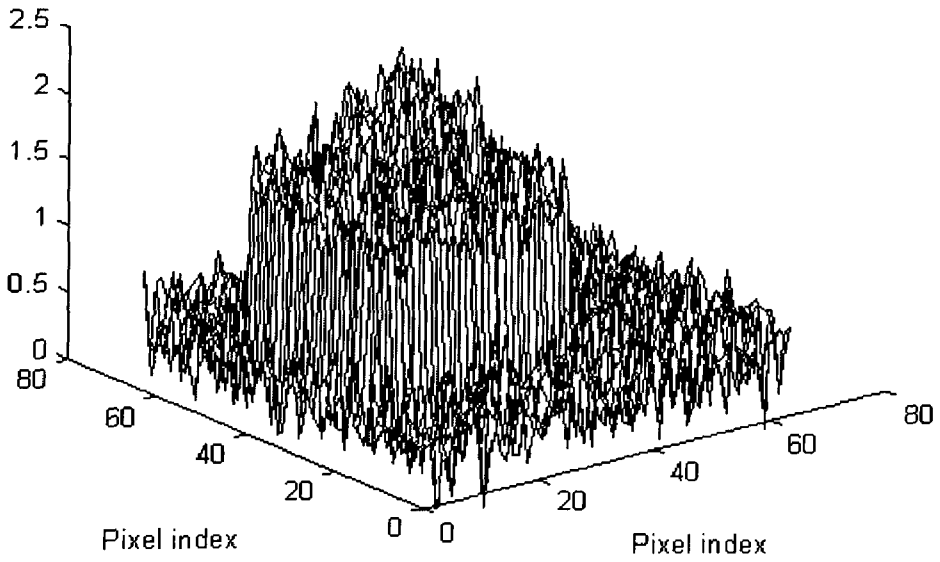

Figure 4.5. "Cake2" corrupted by additive Gaussian noise.

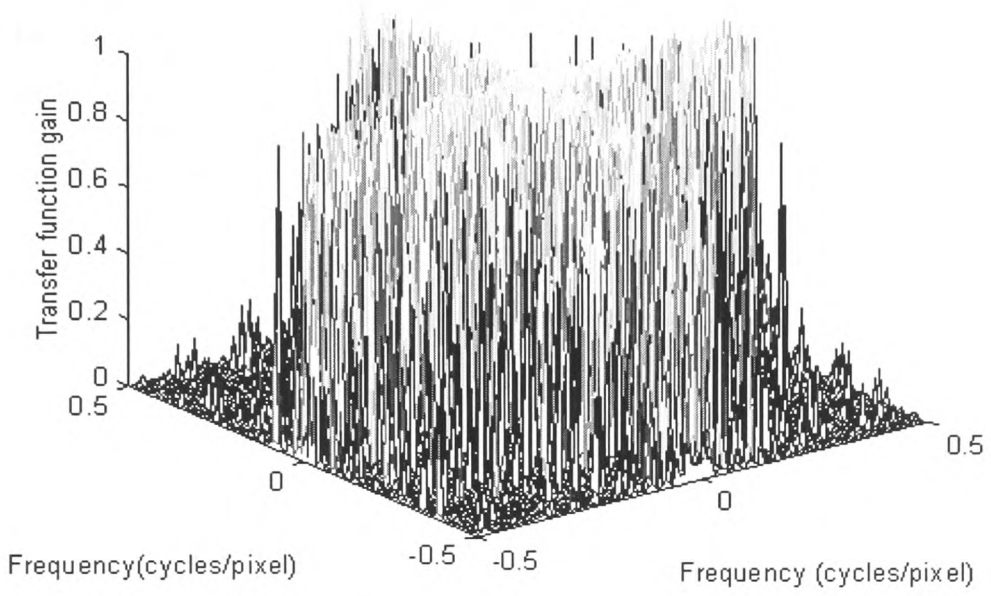

Figure 4.6 Two-dimensional Wiener filter transfer function for example of section 4.3.3

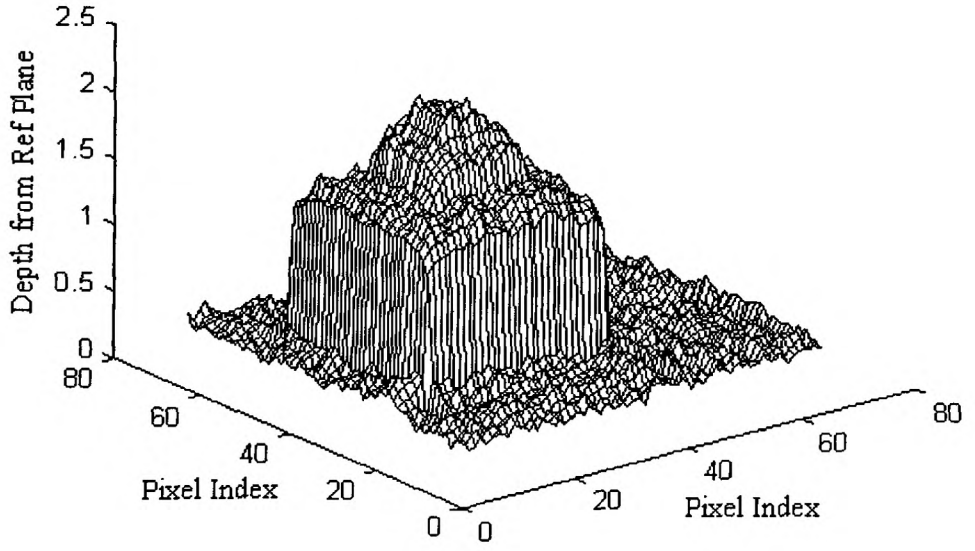

Figure 4.7 Noise-corrupted depth map "Cakes2"after Wiener filtering.

4.4 Robust linear Filters

As discussed in section 4.3.3 for the example of the Wiener filter, linear filters can be designed to be optimal in some sense (minimum mean squared error for the Wiener case) for a given statistical model of the signal and the noise. However if the signal or noise does not satisfy the model then the performance of the filter can be significantly degraded. The techniques of robust filtering seek to design filters that are insensitive to deviations from the assumed model of signal and noise. At the same time robust filters, whilst not optimal for the assumed model of signal and noise process, must perform acceptably for such signals. In practice robust filtering techniques have a defined performance for a range of signal and noise models. A performance criterion that is often used for such filters is the worst case performance for the allowed models of signal and noise. Where a filter is optimal in this performance criterion with respect to a set of allowed models the filter is said to be minmax robust. An expectation for minmax filters is that by optimising the worst case performance the performance for most other cases is not far worse than the optimal performance.

The models of signal and noise are often based on assumptions about their probability distributions and the parameters of the distributions. For example an assumption of Gaussian noise requires the estimation of two parameters, the mean and the variance, of the noise. Model uncertainty can encompass uncertainty in the parameters or uncertainty in the functional form of the distribution, for example whether the distribution function is Gaussian or Laplacian. Robust techniques are concerned with the latter type of model uncertainty. Robust techniques are broadly applied to the two problem areas of signal estimation and signal detection. The area with which this thesis is concerned is that of estimation which is concerned with the problem of predicting or smoothing a signal which has been corrupted by noise using only the corrupted measurements. A review and analysis of robust techniques in both these areas is given by Kassam and Poor (Kassam and Poor,

1985). In this review an example of a Wiener type filter is given in which the ideal output signal to noise ratio versus input signal to noise ratio is compared with the worst case performance of the filter assuming a plausible set of signal and noise spectra over the same range of input signal to noise ratios. It is shown that the optimal filter would produce worse output signal to noise ratio than was presented to its input under these conditions. A performance criterion for robustness for such filters is the ratio of the worst case mean squared error performance over an input set of noise and signal spectra to the optimal performance. A minmax design aims to minimise this ratio. A solution for such robust Wiener filters was obtained for a general case by Poor (Poor, 1980). Poor's solution is to find the least favourable pair of noise and signal spectra $PSD_{X\_least\_favourable}[\omega_k, \omega_l]$ and $PSD_{n\_least\_favourable}[\omega_k, \omega_l]$ (i.e. the spectra which are closest in shape) out of the allowed sets of noise and signal spectra, and then to find the optimal Wiener filter for that pair as in equation 4.10. Robust filters designed using the above approach are discussed in the review paper by Kassam and Poor (*op cit.* Kassam and Poor, 1985) for four different examples of spectral uncertainty in the signal and noise models. The review also discusses optimum and robust filters for the case of noise correlated with the signal.

A reason for deviation from assumed noise statistics (usually Gaussian) and hence the reason for the need for robust filtering techniques is the presence in the noise-corrupted signal of statistical outliers. These outliers are due to gross errors. As has been shown in section 3.4 such outliers are present in the depth map obtained using a correlation-based matcher

4.5 Filtering based on Singular Value Decomposition

A non-linear technique for noise filtering is described in Andrews and Patterson (Andrews and Patterson, 1976) and developed by (Konstantinides *et al.*, 1997) for application to data compression. This technique is based on the Singular Value Decomposition of a matrix. The Singular Value Decomposition is also discussed in another context in section 5.5.2.

The Singular Value filtering technique is based on a theorem from linear algebra (Lawson and Hanson, 1974) that any I by J matrix **B** having I greater than or equal to J can be factored into three matrices **U**, **W**, and **V** such that:

$$\mathbf{B} = \mathbf{U}.\mathbf{W}.\mathbf{V}^T \qquad 4.12$$

Where **U** is I by J, **W** is J by J, and **V** is J by J. The matrix **U** has columns that are orthornormal and therefore is an orthogonal matrix. **V** is also an orthogonal matrix. The matrix **W** has values on the main diagonal only which are also all positive, all off-diagonal values being zero. The elements on the diagonal of **W** are called the 'Singular Values' and are usually arranged in descending order. Any matrix can be decomposed as equation 4.12. The rank of a matrix is the number of its nonzero singular values. Hence the rank r of an m by n matrix representing a noiseless image can be determined. If the image is corrupted by additive Gaussian noise then the noise can be represented by a full rank, R noise matrix of the same dimensions as the original image matrix. The resulting noisy image will now have small but nonzero values for the last R-r singular values. For zero mean Gaussian noise of standard deviation σ, the upper bound on these remaining singular values is given by (Konstantinides and Yao, 1988):

$$\varepsilon \leq \sqrt{mn}.\sigma \tag{4.13}$$

The filtering method is simply to determine the singular values of a noisy image matrix and replace those singular values that are smaller than ε, with zero. The filtered image is then reconstructed using equation 4.12 using the adjusted singular values in matrix **W**.

The algorithm is computationally intensive if applied to the whole image at once. (*op cit.* Konstantinides *et al.*, 1997) apply the singular value decomposition to non-overlapping blocks in the image. They also determine a value for ε using a plot of the second derivative of the compression versus ε for a sample part of the image. The ε that maximises this plot is used as a threshold for the remainder of the image. The algorithm performs well in the tests presented in the paper, however the method is predicated and tested on additive Gaussian noise.

Example

Figures 4.8 To 4.11 show the results of applying the basic Singular Value Decomposition method of filtering to a simulated depth map which has been corrupted by additive Gaussian noise of standard deviation 0.2. The threshold level for ε given by the method of (Konstantinides *et al,* 1997*).* for this depth map is 12.75. The first four singular values of the noisy depth map are: 55.7369, 5.5780, 3.1923, and 2.8919. Accordingly all except the first singular value should be replaced with zeros. However the mean squared error after filtering is best in this case when the first two singular values are retained. The errors obtained are shown in Table 4.1. Also shown in the table are the mean squared errors after applying the method to the depth map corrupted with mixed Gaussian and impulsive noise.

Number of singular values retained for Gaussian noise case	**Resulting Mean squared error after filtering for Gaussian noise case (cf. 0.04 before filtering)**
1	0.0085
2	0.0029
3	0.0054
Number of singular values retained for mixed impulsive and Gaussian noise	**Resulting Mean squared error after filtering for mixed noise (cf. 0.44 before filtering)**
1	0.022
2	0.046
3	0.066

Table 4.1 Summary of performance of SVD filtering as a function of the number of singular values retained for both Gaussian and mixed noise

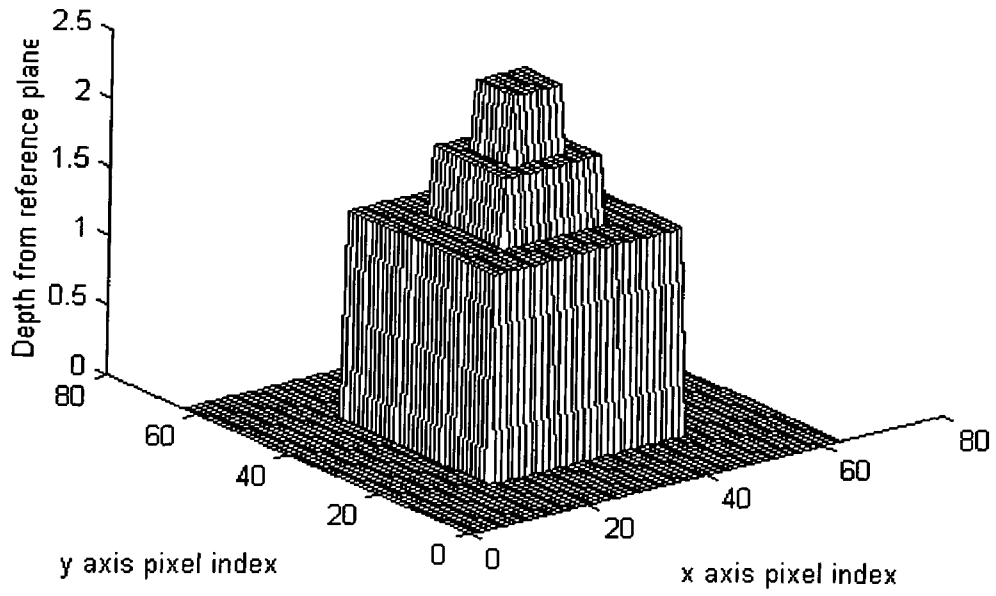

Figure 4.8 Simulated Depth map before filtering using SVD technique

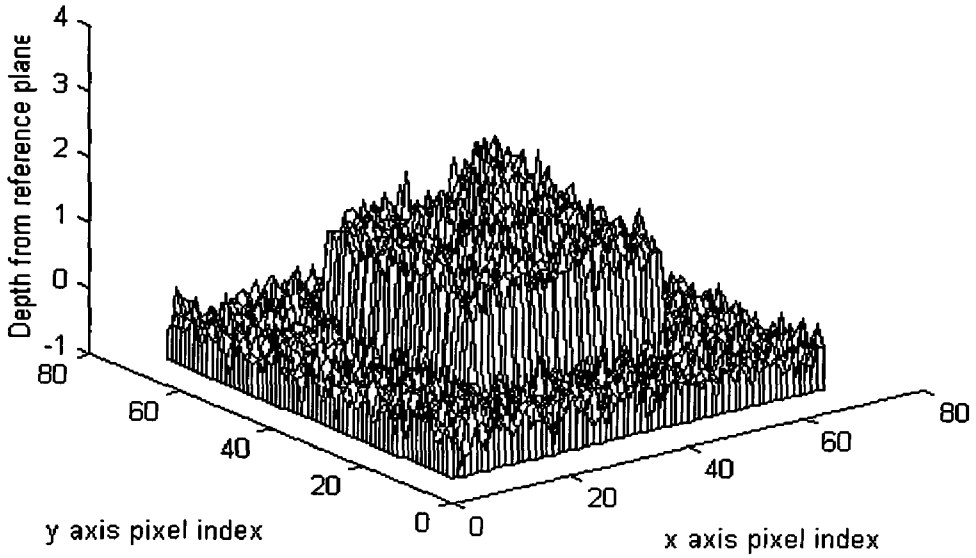

Figure 4.9 Simulated Depth map with added noise before filtering using SVD technique

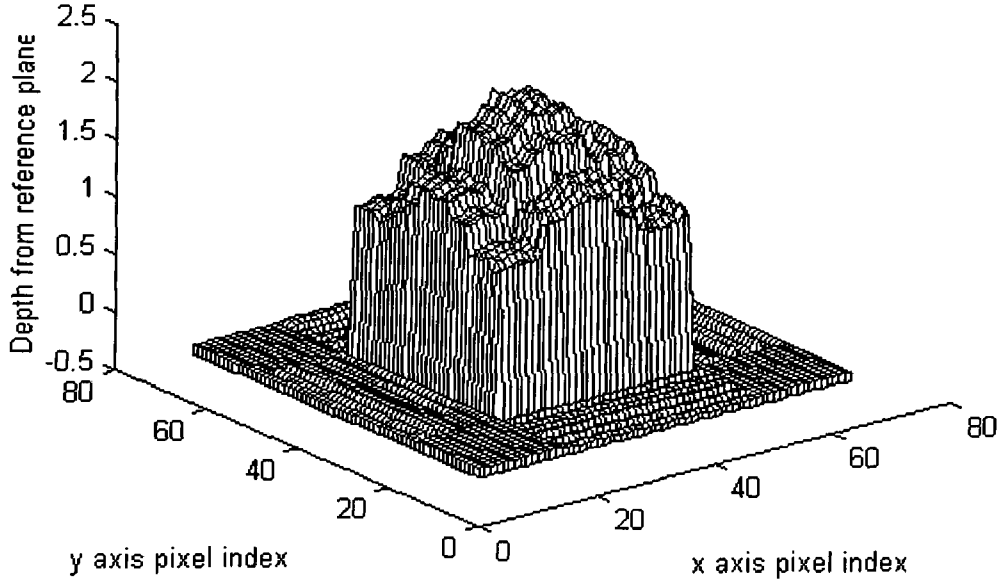

Figure 4.10 Simulated Depth map with added noise after filtering using SVD technique, keeping only the first singular value.

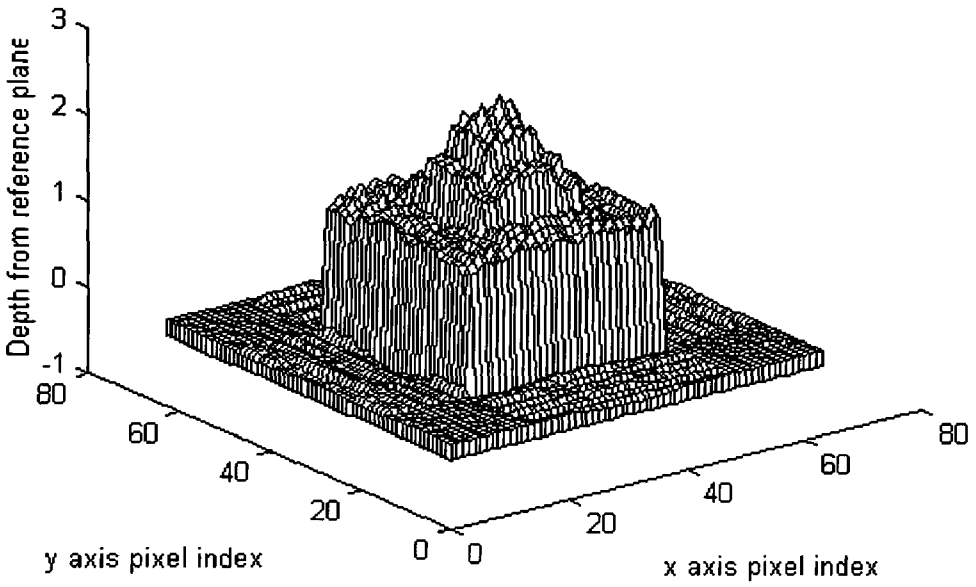

Figure 4.11 Simulated Depth map with added noise after filtering using SVD technique, keeping first two singular values

4.6 Robust Non-linear filters

4.6.1 Median filters

The median of a vector $\mathbf{x}$ of length n, $med(\mathbf{x}_n)$, is computed from the vector $\mathbf{x}_{ranked}$ by:

$$med(\mathbf{x}_n) = \begin{cases} \mathbf{x}_{ranked}\left(\frac{n+1}{2}\right) & \text{if } n \text{ is odd} \\ \frac{\mathbf{x}_{ranked}\left(\frac{n}{2}\right) + \mathbf{x}_{ranked}\left(\frac{n}{2}+1\right)}{2} & \text{if } n \text{ is even} \end{cases} \qquad 4.14$$

where $\mathbf{x}_{ranked}$ is the vector formed by rank ordering the n samples in the vector $\mathbf{x}_n$ and $\mathbf{x}_{ranked}(i)$ is the ith element of the vector. The median can be used as a function operating on the samples within

a filter window to give the median filter. A discussion of the theoretical properties of the median filter is given in chapter 4 of (Pittas and Venetsanopoulos, 1990). The median filter is an example of M estimators of location (Huber 1964), for which:

$$Loc(\mathbf{x}) = \text{the value of } u \text{ which minimises: } \sum_{i=1}^{i=n} L(x_i - u) \qquad 4.15$$

where the function $L(x) = x^2$ for a sample mean, $L(x) = |x|$ for the median. For additive noise of probability density function $f(x)$ if $L(x) = -\log(f(x))$, the M estimator of location is the maximum likelihood estimator (*op cit* Pittas and Venetsanopoulos, 1990), (*op cit.* Kassam and Poor, 1985).

Quantitative methods of comparing the performance of filters are the use of influence functions, asymptotic variance, and asymptotic relative efficiency. The influence function measures the effect of a single outlier on the estimation of location provided by a filtering scheme (*op cit* Pittas and Venetsanopoulos, 1990). For some estimator of location, T(**x**), where **x** is a vector of observations with idealised distribution F, the Influence Function, $IF(x;T,F)$, is given by:

$$IF(x;T,F) = \lim_{t\to 0} \frac{T((1-t)F + t\delta x) - T(F)}{t} \qquad 4.16$$

Where it is assumed that the estimator T(.) correctly measures the location parameter asymptotically for the true distribution F. Equation 4.15 can also be used on an empirically measured distribution (histogram) of n-1 observations G_{n-1} by substituting G_{n-1} for F and $t = \frac{1}{n}$, taking the limit as being the value for large n. The highest value of the Indicator Function is taken as the "gross-error sensitivity". An estimator with a finite gross-error sensitivity is said to be B robust. (*op cit* Pittas and Venetsanopoulos, 1990)derive the influence function of the median and

arithmetic mean for Gaussian distributed samples From the plot of figure 4.12, reproduced from (*op cit* Pittas and Venetsanopoulos, 1990) it can be seen that the arithmetic mean is not B-robust in contrast to the median, showing that the median produces an estimate of location which is stable to outliers in the data.

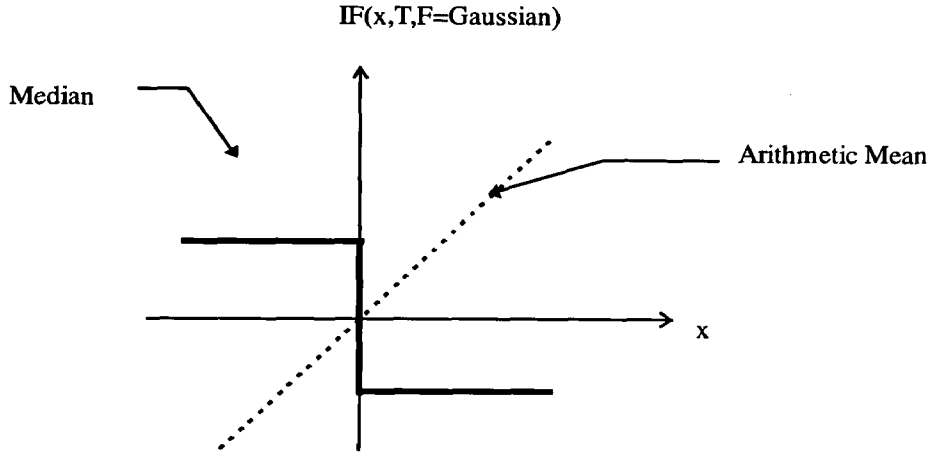

Figure 4.12: Plots of influence functions of mean and median for Gaussian distributed data reproduced from (Pittas and Venetsanopoulos, 1990).

A quantity called the Asymptotic Variance V(T,F) can be derived from the Influence Function for the estimator T (*op cit* Pittas and Venetsanopoulos, 1990) given by:

$$V(T,F)=\int IF(x;T,F)^2 dF \qquad 4.17$$

From the asymptotic variance a comparison of two estimators T and S can be made using the asymptotic relative efficiency, ARE, given by:

$$ARE(T,S)=\frac{V(S,F)}{V(T,F)} \qquad (4.16)$$

Comparing the asymptotic relative efficiency for the arithmetic mean as S and the median as T for various distributions gives an asymptotic relative efficiency of less than unity for distributions which are short tailed such as the Gaussian, implying that the mean is a better estimator of locations for these distributions. An asymptotic relative efficiency of greater than unity is given for

thicker tailed distributions such as the Laplacian distribution, implying that the median is a better locator for such distributions. For mixed distributions, such as the mixed Gaussian and impulsive distribution characteristic obtained for the noisy disparity maps discussed in Chapter 3, which of the median or mean filter is the more efficient depends on the balance between the Gaussian and impulsive components of noise. Since this will vary from one part of the disparity map to another, an ideally efficient (at least in an asymptotic relative efficiency sense) filter will change its behaviour depending on the noise statistics.

Example

Figure 4.13 Shows the Gaussian noise-corrupted simulated depth map of figure 4.5 after filtering with a 3 x 3 window Median filter. The mean square error after filtering in this case was 0.0089.

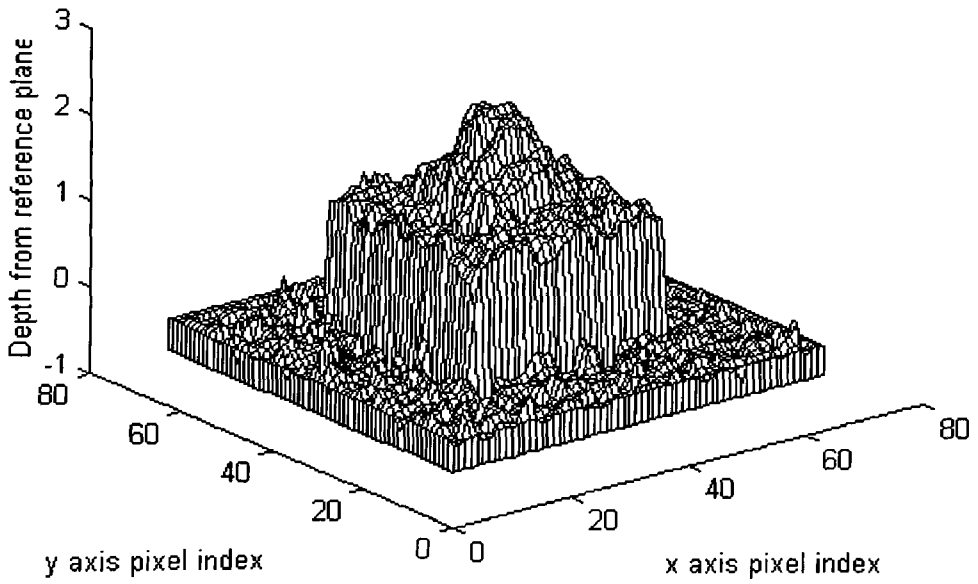

Figure 4.13 Gaussian noise-corrupted depth map of figure 4.5, after filtering with a 3 by 3 median filter

4.6.2 Other Non-linear Filters

Other, related, types of non-linear filters are alpha and modified trimmed mean filters (Lee and Kassam, 1985), truncated median (mode) filters (Evans and Nixon, 1995) weighted order statistic (WOS) filters (Kleihorst *et al.*., 1997), filters that are based on morphological operations (Pessoa and Maragos, 1998)(Singh and Siddiqi, 1997), hybrid FIR/WOS filters (Heinonen and Neuvo, 1987), and filters based on neural networks (Yin *et al.*. 1991), (Haykin *et al.*., 1997).

4.7 Filters based on statistical *a priori* models and regularisation

This class of filters has been used to restore images affected by degradation effects such as blurring, noise and non-linear transformation. The approach was first described in the seminal paper of (*op cit.* Geman and Geman, 1984) and placed in the general framework of the idea of the regularisation of ill-posed problems by (Poggio *et al.*.1985). These filters are more often described as stochastic restoration techniques rather than filters. Nevertheless they can be looked on from the point of view of a Black Box filtering process. However, for these filters the processes that take place inside the Black Box are iterative processes which converge toward the final output. The approach used is based on maximising the sum of the conditional probability of the original corrupted image given a possible solution and the probability of the trial solution as a function of the trial solution. Since the theoretical end point maximises these probabilities (which is assumed to give a unique solution), then the process in the Black Box is a unique mapping and satisfies the requirement outlined in section 4.2 for the process to be a mathematical function. However since there are many local maxima to the probability distribution as a function of possible solutions, and since a stochastic Monte Carlo approach is generally taken to obtaining the solution, the solution may not in practice be unique for a given input. In this case the mapping process taking place

within the Black Box will not be a 'function' in the strict mathematical sense. Nevertheless the ideal mapping process can be regarded as a function.

In the work of (*op cit.* Geman and Geman, 1984) images degraded by blurring, non-linear distortion and noise were restored by choosing the estimate for the original image which maximised the *a posteriori* probability of that estimate conditioned on the probability of that estimate given the observed degraded image. The original image and the estimate were each modelled as a pair of Markov Random Fields (MRFs), consisting of a matrix representing the pixel grey level intensities and another matrix representing line (edge) processes. MRFs are an extension into two-dimensions of the idea of a Markov chain. In an MRF the conditional probability of a particular pixel taking a particular value given the values of all the pixels in the remainder of the image is the same as the conditional probability of the pixel taking the same value given the values of the pixels in a defined neighbourhood known as a clique. The size and shape of the clique is a parameter of the MRF. MRFs capture the local correlation of most real images. An MRF can also be identified with a corresponding Gibbs probability distribution. The probability distribution allows a computationally more tractable description of the MRF representing the image. This is because the probability measure of each pixel in the neighbourhood taking a given value is expressed in terms of potentials rather than conditional probabilities. The sum of these potentials over the clique gives an energy function for a given configuration. This energy function is arrived at by consideration of an appropriate image corruption model of blurring, non-linear transformation, and additive or multiplicative independent noise. Maximising the posterior distribution is equivalent to minimising the energy function. This minimisation is carried out in the paper using the Metropolis algorithm, (Metropolis *et al.*1953) which is itself discussed further in a different context in Chapter 5 of this thesis. This is a Monte Carlo technique but with optimised sampling of the system configuration.

Because the energy function is only computed over the local clique the algorithm readily allows parallel computation.

It was pointed out in (*op cit.* Poggio *et al.*1985) and discussed in (Bertero *et al.*1988) that the technique of Geman and Geman is an example of a stochastic approach to the regularisation of an ill-posed problem. Such problems were discussed in section 3.6. In this case the *a priori* knowledge is captured in the choice of probability distribution for the pixel values or equivalently the potential function of the Markov random field. More recent papers adopting the general approach of Geman and Geman either view the technique from the standpoint of the regularisation of an ill-posed problem or tend to emphasise the Bayesian nature of the approach. The choice of the potential function is critical to the effect of the method on the preservation of edges in the original images. As discussed in section 3.6 for the deterministic minimisation of a potential function, a simple quadratic potential leads to the loss of edges in the original image. This problem is avoided in Geman and Geman by the use of the line process matrix, but leads to a non-quadratic potential. In general edge preservation demands that the potentials to be minimised become non-convex (Charbonnier *et al.* 1997). (Blake and Zisserman, 1987) in their Graduated Non-Convexity algorithm, reduce the problem to the problem of minimising a sequence of energy functionals, each of which can be minimised deterministically.

4.8 Summary of chapter 4

This chapter has carried out a brief review of possible filtering strategies that could be adopted to filter the noisy depth maps produced by the SSD stereo matcher. From Chapter 3 it has been established that the depth map noise is not purely Gaussian, but that it contains a substantial

impulsive noise component. The chapter began by describing the class of linear filters, focusing on FIR filters, and leading to a discussion of the optimal Wiener filter. For a frontoparallel flat surface corrupted by Gaussian noise, the optimum filter reduces to an averaging filter, which can be reasonably approximated by a moving average filter. However, the moving average, or indeed a Wiener filter predicated on Gaussian noise, is not robust to outliers such as exist in the presence of impulsive noise.

Non-linear filters are then discussed starting with an example of a filter based on the singular value decomposition of an image matrix. The chapter then discussed the median filter, which is more robust in the presence of outliers and also preserves edges and discontinuities. However, the median filter is not optimal for Gaussian noise. Other non-linear filtering approaches that have been proposed in the literature were then briefly enumerated. Finally the chapter discussed the widely explored iterative Bayesian approach to image restoration based on a Markov random field statistical model of images and their discontinuities. The median, Wiener, and moving average filters will be used as benchmarks with which the new fuzzy filters proposed in this thesis will be compared in Chapter 7.

4.9 References

(Andrews and Patterson, 1976) Andrews H.C. and Patterson C.L. "Singular Value Decompositions and Digital Image Processing." IEEE Transactions on Acoustics, Speech, and Signal Processing Vol. 24 pp 26-53. Feb 1976.

(Bertero *et al.*1988) Bertero, M. Poggio, T. and Torre V. "Ill Posed Problems in Early Vision." Proceedings of the IEEE Vol. 76 No. 8 pp 869 - 889, Aug. 1988

(Blake and Zisserman, 1987) Blake A. and Zisserman A. "Visual Reconstruction." MIT Press 1987.

(Brainerd *et al.*, 1967) Brainerd B., Clarke D.A., Liebovitz M.J., Ross R.A , and Scroggie G.A. "Functions." Frederick Warne. 1967.

(Charbonnier *et al.* 1997) Charbonnier P., Blanc-Feraud L., Aubert G., and Barlaud M. "Deterministic Edge-Preserving Regularisation in Computed Imaging." IEEE Transactions on Image Processing Vol 6. No 2 pp 298-311. Feb 1997.

(Evans and Nixon, 1995) Evans A.N. and Nixon M.S. "Mode Filtering to reduce ultrasound speckle for feature extraction." IEE Proceedings Vision and Image Signal Processing. Vol 142 No 2. pp 87-94. April 1995.

(Geman and Geman, 1984) Geman S. and Geman D. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 6 No. 6.

(Haykin *et al.*,1997) Haykin S., Yee P., and Derbez E. "Optimum Nonlinear Filtering." IEEE Transactions on Signal Processing Vol 45 No 11, pp 2774-2786. November 1997.

(Heinonen and Neuvo, 1987) Heinonen P. and Neuvo Y. "FIR-Median Hybrid Filters." IEEE Transactions on Acoustics Speech and Signal Processing. Vol 35 No 6 pp 832-838. June 1987.

(Huber 1964) Huber P.J. "Robust estimation of a location parameter." Annals of Mathematical Statistics Vol 35 pp73-104 1964.

(Kassam and Poor, 1985) Kassam S.A.and Poor H.V. "Robust Techniques for Signal Processing: A Survey." Proceedings of the IEEE Vol. 73 No.3 March 1985

(Kleihorst *et al.*, 1997) Kleihorst R.P., Lagendijk R.L. Biemond J. "An Adaptive Order Statistic Noise Filter for Gamma-Corrected Image Sequences." IEEE Transactions on Image Processing Vol. 6 No. 10 pp 1442-1446 October 1997.

(Konstantinides and Yao, 1988) Konstantinides K. and Yao K. "Statistical Analysis of Effective Singular Values in Matrix Rank Determination." IEEE Transactions on Acoustics Speech and Image Processing, pp757-763, May 1988.

(Konstantinides *et al.*, 1997) Konstantinides K, Natarajan B., and Yovanof G.S. "Noise Estimation and Filtering Using Block Based Singular Value Decomposition." IEEE Transactions on Image Processing. Vol. 6 No. 3, March 1997.

(Kosko, 1992) Kosko B. "Fuzzy Systems as Universal Approximators." IEEE International Conference on Fuzzy Systems San Diego. pp 1153-1162. 1992.

(Kosko, 1994) Kosko B "Fuzzy systems as universal approximators." IEEE Transactions on Computing Vol. 43 pp 1329-1333. 1994.

(Lawson and Hanson, 1974) Lawson C.L. and Hanson R.J. "Solving Least Squares Problems." Prentice-Hall. 1974.

(Lee and Kassam, 1985) Lee Y.H. and Kassam S.A. "Generalised Median Filtering and Related Nonlinear Filtering Techniques." IEEE Transactions on Acoustics Speech and Signal Processing. Vol 33 No 3 pp 672-683 June 1985

(Li, 1994) Li Sz. "Markov Random Field Models in Computer Vision." Lecture notes in Computer Science: Proceedings of the Third European Conference on Computer Vision. Springer-Verlag. May 1994.

(Metropolis *et al.*, 1953) Metropolis N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H., Teller E. "Equation of State Calculations by Fast Computing Machines" Journal of Chemical Physics Vol 21, No 6. pp 1087-1092. June 1953.

(Pessoa and Maragos, 1998) Pessoa L.F.C. and Maragos P. "MRL Filters: A General Class of Nonlinear Systems and Their Optimal Design for Image Processing." IEEE Transactions on Image Processing Vol. 7 No. 7 pp 966-978 July 1998

(Pittas and Venetsanopoulos, 1990). Pittas I. and Venetsanopoulos A.N. "Nonlinear Digital Filters Principles and Applications." 1st Edition. Kluwer 1990.

(Poggio *et al.*1985) Poggio T. Torre V. Koch C. "Computational vision and regularisation theory." Nature Vol. 317. Sept 1985.

(Poor, 1980). Poor H.V. "On Robust Wiener Filtering." IEEE Transactions on Automatic Control Vol. 25 pp 531-536. June 1980.

(Singh and Siddiqi, 1997) Singh B. and Siddiqi M.U. "MAP Estimation of Finite Gray-Scale Digital Images Corrupted by Supremum/Infimum Noise." IEEE Transactions on Image Processing Vol. 6 No. 8 pp 1077-1088 August 1997

(Wang, 1992) Wang L-X. "Fuzzy Systems are Universal Approximators." IEEE International Conference on Fuzzy Systems San Diego pp 1163-1169. 1992

(Wiener, 1949) Wiener N. "Extrapolation, Interpolation, and Smoothing of Stationary Time Series, with Engineering Applications." MIT Press 1949.

(Yin *et al.* 1991) Yin L. Astola J., Neuvo Y. "Adaptive Neural Filters." Proceedings of the IEEE Workshop on Neural Networks for Signal Processing. pp. 503-512. 1991.

Chapter 5 : Sugeno Fuzzy Systems and their Training.

5.1 Introduction

This chapter describes the fuzzy inferencing systems which are used in this thesis as the basis for filters to smooth signals corrupted by mixed Gaussian and impulsive noise such as the noisy depth maps obtained using correlation-based matching described in Chapter 3. After a description of the fuzzy inferencing systems used to realise such filters, this chapter seeks to outline the contribution made by the thesis to the training of fuzzy inferencing systems using various methods based on the simulated annealing algorithm of (Metropolis *et al.*1953).

The training approach described in this chapter allows the automatic adjustment of all the free parameters of the fuzzy inferencing systems so that they can perform an approximation to the underlying function mapping exemplar input data to its corresponding output data. The chapter also describes the application software called 'FTEST' that was developed to implement the training algorithms. This FTEST software is used to create and train an instantiation of a fuzzy inferencing system object. This object has the property of persistence as it can be stored to hard disk and recreated for use as a filter in another software application called 'WINIM' developed to perform the image processing necessary to perform correlation-based matching and depth map re-creation.

5.2 Fuzzy Inferencing Systems

This section forms a brief outline of the development and theoretical background of fuzzy systems and a comparison of the Mamdani and Sugeno approaches to fuzzy inferencing systems.

5.2.1 Fuzzy systems in engineering

The use of fuzzy inferencing systems for engineering problems was originally conceived by Lotfi Zadeh (Zadeh, 1965), (Zadeh, 1971), and (Zadeh, 1973). Zadeh built on the mathematical foundations of multivalued logic to allow the mathematical and hence the computational manipulation of variables such as speeds whose values can be expressed as linguistic terms such as 'fast' or 'slow'. These linguistic terms capture the uncertainty inherent in much of 'real-world' reasoning. Fuzzy inferencing systems attempt to emulate the flexibility and robustness with which the human mind manipulates such linguistic variables to produce sensible working results in the face of uncertainty. An obvious example of the robustness of human reasoning processes is the ability of humans to control systems under grossly varying conditions. Moreover, what is significant in the context of fuzzy inferencing is that human operators can describe their control actions and the conditions giving rise to them in linguistic terms such that another human can quickly learn the complex control actions. It is natural, then, that the earliest, and still the commonest, application area for fuzzy systems is in control engineering. The seminal papers in this field were those of (Mamdani, 1974), and (Mamdani and Assillian, 1975), in which an experimental boiler and steam engine system was controlled. The Mamdani approach is to model the relationship between input fuzzy linguistic variables and output fuzzy linguistic variables using a set of control rules to generate a fuzzy relation. An essentially equivalent approach is that of (Holmblad and Østergaard, 1982) which considers the degree of activation of a number of rules in parallel. The effective equivalence of these two approaches is discussed in some detail (Bandemer and Gottwald, 1995). Both for convenience, and because of the effective equivalence between them, both approaches will, from this point in the thesis, be referred-to as Mamdani inferencing systems. The Mamdani approach to control and modelling is now widely used as a standard approach in the research literature. It is also the predominant inferencing system used in fuzzy logic-based filters described in the published

literature on fuzzy filters which is reviewed in Chapter 6 of this thesis. Mamdani Inferencing is discussed in more detail in section 5.3.3

5.2.2 Fuzzy sets and fuzzy inferencing - a brief overview

Fuzzy systems are based on an extension of crisp set theory. In normal crisp sets an element either is or is not a member of a set. In crisp sets the degree of membership of an element in a given set can only have the values '1' or '0' when the element is or is not a member of that set. In fuzzy sets an element can have any degree of membership in the range [0,1] in a given set. This captures the imprecision inherent in linguistic variables such as 'tall', 'fast' or 'large'. A crisp variable such as a speed in mph can be assigned a degree of membership in a set called 'fast' according to a function of the crisp input value, called a membership function for the set (denoted by $\mu_A(x)$ for the set A and the argument x) The range of values of the argument for which the membership value is non-zero is called the support of the fuzzy set and the set of all possible values for the argument x is called the universe of discourse for the set A. Membership functions are often chosen to be Gaussian, trapezoidal, or triangular in shape, the peak of the function at a membership degree of '1' having the argument representing the most typical value of the fuzzy set in question. A fuzzy singleton is a special type of membership function, which has a non-zero value for only one argument – i.e. it is a spike situated at that argument value.

The normal set operations of intersection, union, and complement can be extended to fuzzy sets by applying suitable operators to the membership functions defining the fuzzy sets. Operators suitable for carrying out fuzzy intersection are called t-norms (represented in general by $\bigstar$) and those operators suitable for fuzzy union are called t-conorms (represented by $\oplus$). In many engineering applications the *min()* operator is used for fuzzy intersection and the *max()* operator

for fuzzy union. The membership function for the complement of a fuzzy set A is typically defined by the operation $1-\mu_A(x)$.

In crisp set theory a (binary) relation between two sets of objects is defined so that given an ordered pair (x,y) in the Cartesian product of two sets A ($x \in$ A) and B ($y \in$ B), x is related to y by the relation R if, and only if, (x,y) is in R (Epp 1990). The relation is itself a set and expresses some relationship between the elements of the sets A and B. Similarly a fuzzy relation R can be defined for two fuzzy sets A and B defined by $\mu_A(x)$ and $\mu_B(y)$ such that the relation is defined by a membership function $\mu_R(x,y)$. If the universe of discourse of fuzzy set A is U and that of B is V then the relation has a universe of discourse of $(U \; X \; V)$. The fuzzy relation is itself a fuzzy set and fuzzy relations can therefore be subjected to the operations of intersection, union, and complement. The intersection and union of two relations performed using t-norm and conorm operators are called compositions of the two relations. Thus the composition which is the union of two relations R(U,V) and S(U,V) is given by $\mu_R(x\ y) \oplus \mu_S(x,y)$. Compositions can also be defined, again by analogy with crisp set theory, for two fuzzy relations e.g. $R(U,V)$ and $S(V,W)$ that are defined on different product spaces, provided that they share one common universe of discourse. In this case if $z \in W$ then the composition $C=R_oS$ which is the union of R and S is defined by:

$$\mu_{RoS}(x,z) = sup_y\ (\mu_R(x,y) \bigstar \mu_S(y,z).) \qquad 5.1$$

Where the *sup* operator is the supremum of the membership function and the $\bigstar$ operation is a t-norm.

The significance of the ideas of relations and compositions is that by using compositions of fuzzy relations, complex relationships between fuzzy sets can be described. In effect the composition of two relations is equivalent to chaining the two relations. Moreover the composition of a fuzzy set F defined by $\mu_F(x)$ with a relation R defined by $\mu_R(x,y$ using equation

5.1 gives the (fuzzy) output when a fuzzy set excites a fuzzy relation. In this case equation 5.1 reduces to:

$$\mu_{FoR}(y) = sup_{x} (\mu_F(x) \star \mu_R(x,y).) \qquad 5.2$$

The process of making a logical inference is a process of arriving at a conclusion from a hypothesis. In conventional logic the process of inferring an output value from the values of input variables proceeds by a statement or rule of the form:

If *input variable 1 is set a* **AND** *input variable 2 is set b* ... **AND** *input n is set z* **THEN** *output variable 1 is output set oa* .

The first part of the statement or hypothesis up to the **THEN** is called the antecedent and the second part after the **THEN** is the consequent or conclusion. The antecedent consists of propositions or predicate clauses – e.g. **IF** *input variable 1 is set a* – linked by connectives (**AND** or **OR**). The propositions are either true or false – corresponding to the variable either being or not being a member of the input set named in the proposition. This mode of inferencing is called 'Modus Ponens' and consists of the following steps:

x is a member of set A

IF x is a member of set A THEN y is a member of set B

y is a member of set B.

The second step is an implication- A implies B or $A \rightarrow B$

This mode of inferencing can be extended (Generalised Modus Ponens, GMP) using the idea of fuzzy sets to read:

x is a member of set A to some degree

IF x is a member of set A THEN y is a member of set B

y is a member of set B to some degree.

The fuzzy implication $A \rightarrow B$ is equivalent to a fuzzy relation and is described by a membership function $\mu_{A \rightarrow B}(x,y)$. The output fuzzy set resulting from GMP is given from equation 5.2 by

$\mu_B(y) = sup_x (\mu_A(x) \star \mu_{A\rightarrow B}(x,y))$. This is called the compositional rule of inference (*op cit* Zadeh, 1973).

In order to apply GMP and equation 5.2 to an engineering problem involving the determination of some output value from multiple input values, the following issues have to be addressed:

a) A method of transforming the usually crisp numerical input data into an input fuzzy set $\mu_A(x)$. This is called fuzzification.

b) An operator has to be chosen for combining multiple input sets through the AND connective.

c) A suitable t-norm must be chosen for the $\star$ operation.

d) A suitable method is required for determining the relation defined by $\mu_{A\rightarrow B}(x,y)$ (the implication operator)

e) In general many rules will be needed, each expressed using an equation of the form of 5.2, thus some means is needed to combine the resulting multiple fuzzy output sets for a given input

f) The fuzzy output set resulting from a given rule (or combined rules) and input will in general need to be converted to a single crisp value. This process is called de-fuzzification.

The following represent typical choices in engineering systems for the issues a) to f):

a) The input fuzzy set represented by $\mu_A(x)$ in equation 5.2 is derived from the crisp numerical input data by determining the value τ of the membership function for that input data. The input set $\mu_A(x)$ then becomes a fuzzy singleton of height τ. This is called the singleton fuzzifier. The effect of this is to remove the need for the *sup* operator in equation 5.2 since the value of the supremum is simply τ

b) Either the *min* or product operator is typically used for the connective AND.

c) Either the min or product operator is also typically used for the $\star$ operator.

d) The many possible choices of implication operator are reviewed in (Lee, 1990(b)). However (Mendel, 1995) points out there that only certain implication operators satisfy the engineering requirement of preserving causality. The commonly made choices of the $min(\mu_A(x),\mu_B(x))$ (Mamdani, 1974) and $\mu_A(x)\mu_B(x)$ (product) (Larsen, 1980) operators for implication preserve causality, but do not satisfy the truth table for implication in conventional logic. (*op cit* Mendel, 1995) proposes that these operators be called "...engineering implications."

e) The t-conorm – corresponding to fuzzy union- is a generally used method of combining, the final output sets (*op cit* Mendel, 1995)

f) A range of different methods exist for de-fuzzification (*op cit* Lee, 1990(b)), the salient methods being the Max criterion, the mean of maximum and the centre of gravity method. The need for de-fuzzification can be avoided (see section 5.2.3) by using a Takagi-Sugeno inferencing system (Takagi and Sugeno, 1983).

5.2.3 Mamdani and Sugeno Inferencing

The Mamdani inferencing system uses the singleton fuzzifier, the compositional rule of inference using a relation derived from a set of n IF.....THEN rules and one of the defuzzification techniques referred to in section 5.2.2. The overall fuzzy relation derived from the rule base is derived by a composition of all the relations associated with each rule in the rule base. Each relation in the rule base is derived using the *min* implication operator. A general block diagram of a Mamdani inferencing system is shown in figure 5.1

An alternative to the Mamdani approach, but one that is closely related (*op cit* Bandemer and Gottwald, 1995) is to use the idea of a degree of activation for each rule (*op cit* Holmblad and Østergaard, 1982). The degree of activation of each rule is a measure of the degree to which the input value satisfies the input set specified in the rule. This activation degree is then used to

weight the output set associated with that rule and the overall system output is a superposition of all the weighted output sets.

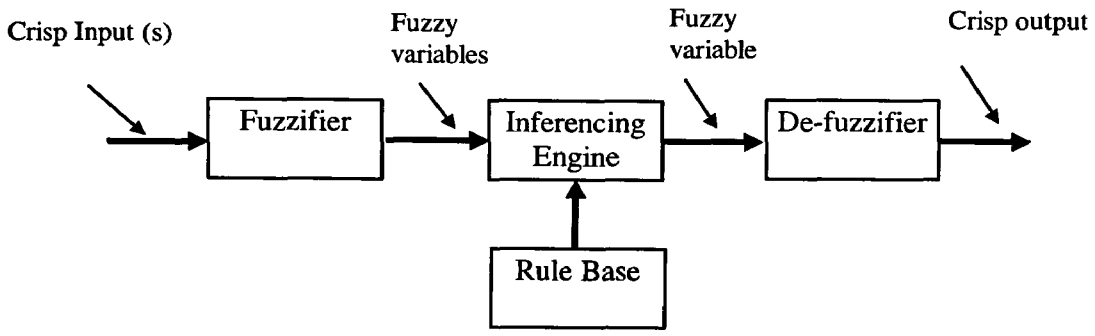

Figure 5.1: General block diagram of a Mamdani fuzzy inferencing system

A different inferencing system to the Mamdani approach is that due to (*op cit* Takagi and Sugeno, 1983). This Sugeno approach avoids the need for a defuzzification stage. The inputs to a Sugeno fuzzy system are crisp numbers which are fuzzified using the fuzzy singleton fuzzifier. A degree of activation, which is a crisp number, is determined for each rule from the degrees of membership of the input values in the fuzzy sets that are the antecedents in the rules.

Up to this stage the Sugeno approach is very similar to the Mamdani approach. The Sugeno system differs, however, in that the output consequent for each rule is a function of the crisp inputs. The function is often either simply a constant associated with each rule (a zero order Sugeno system) or a linear combination of the crisp input values (a first order Sugeno system). However other functions can be used, for example the median of the input values. In this thesis the output functions are referred-to as the 'output sets'. The overall output of the Sugeno system aggregated over all the rules is given by the weighted sum of the output functions where the weighting for each rule's output is equal to the degree of activation of that rule normalised by the sum of the activation degrees over all rules. The zero order Sugeno system is very

similar to a Mamdani type system in which the output sets are fuzzy singletons, and in many cases the result from such a Sugeno system is very close to that which would be obtained using a Mamdani system with centre of gravity de-fuzzification. The general block diagram for a Sugeno system is shown in figure 5.2. Since all the fuzzy systems used in this thesis are Sugeno systems, they are described in more detail in section 5.4.

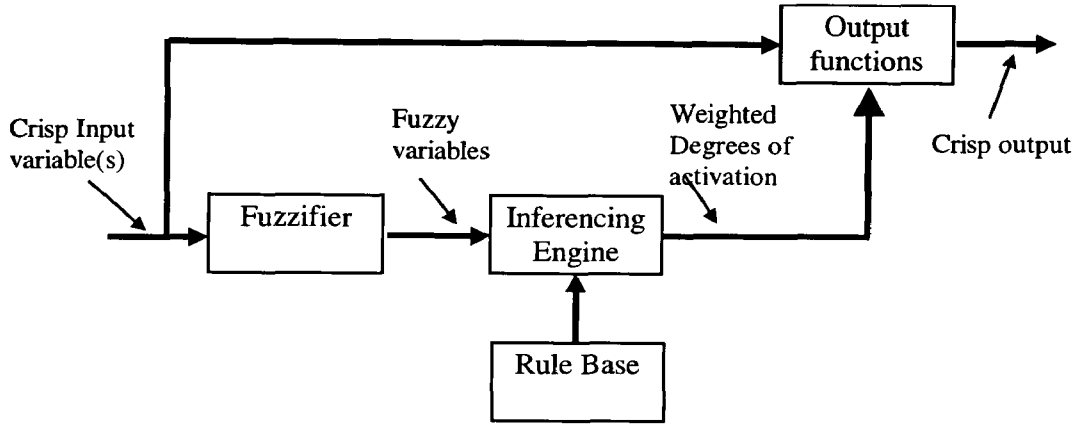

Figure 5.2: General block diagram of a Sugeno fuzzy inferencing system

5.3 Application software 'FTEST'

5.3.1 Purpose of FTEST

Since one of the aims of the work described in this thesis was to investigate the use of fuzzy systems as a basis for non-linear filters, it was necessary to generate the fuzzy systems in software. Moreover it was necessary that the fuzzy systems thus generated could be applied to the task of image processing. The approach taken to tackling these requirements was to create two software applications running under the Microsoft Windows NT ™ operating system using the Microsoft Visual C++ ™ compiler. The first application called FTEST is a tool for creating, configuring and training Sugeno fuzzy inferencing systems. The fuzzy systems created by FTEST can be saved to disk in a format which can be read by the second

application. The second software application, called WINIM, performs the image processing necessary to extract depth and disparity maps from pairs or sequences of two-dimensional images. The fuzzy systems created by FTEST can therefore be used by WINIM to filter noisy depth maps. Sections 5.3.2 to 5.3.4 briefly describe the user interfaces of FTEST.

5.3.2 Fuzzy system editing screen

This screen enables the user to set the overall configuration of the fuzzy system. The configuration of the system includes the definition of the system as a zero order or first order Sugeno system, or the special case of a first order Sugeno system called 'FIR Type'. The FIR type has output set functions that are linear combinations of the crisp input values but which do not include the constant term that is usually included in the linear combination used in first order Sugeno systems. This enables each output set to be equivalent to an FIR filter acting on the input data vector.

The user must also define the length of the input data vector to the fuzzy system in the fuzzy system editing screen through the 'No of inputs' window. The Universe of discourse for the inputs must also be defined. This is the same for all inputs in FTEST since the intended inputs to the fuzzy systems are the depth map samples at each pixel. The specified Universe of discourse is used to generate an out of bounds penalty during training (see section 5.7.4) and is also used to initialise the input sets during automatic rule generation.

The system editing screen also allows a check for the internal consistency of the fuzzy system. This ensures that input sets and output sets referenced in defined rules exist, since fuzzy systems can be saved to disk in a partially completed state.

Figure 5.3 shows the fuzzy system entry screen for FTEST.

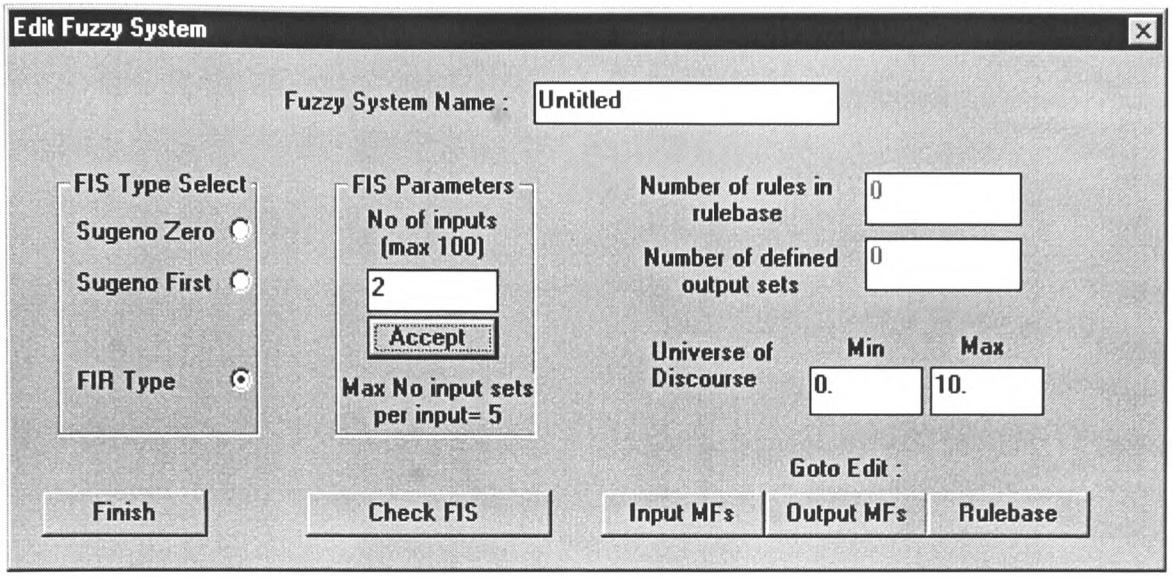

Figure 5.3: Fuzzy system entry screen for FTEST

5.3.3 Membership function editing screen.

This screen allows the user to define the membership function parameters for each the input fuzzy sets. The fuzzy sets can be Gaussian or trapezoidal in shape, with triangular forming a special case of trapezoidal. This is termed the 'MF Flavour' in the editing screen. The membership function can be of 'type' normal, left or right. These are illustrated in figure 5.4 for a trapezoidal membership function. Also shown in figure 5.4 are the parameters for the different trapezoidal membership functions. A left or right trapezoidal membership function requires two parameters to be specified (shown as a, b and g, h in figure 5.4), whereas a normal trapezoidal membership function needs four parameters (shown as c, d, e, f in figure 5.4). All Gaussian membership functions are specified by two parameters, a width parameter and a position parameter, equivalent to the standard deviation and mean for a Gaussian function.

The membership function editing screen is shown in figure 5.5.

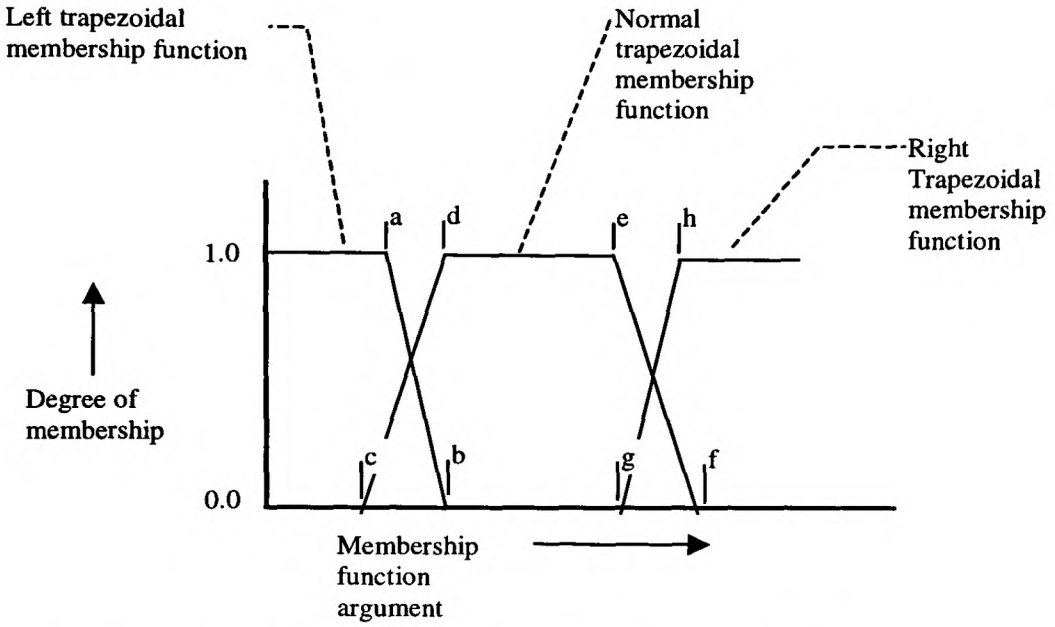

Figure 5.4: Illustration of different types of trapezoidal membership functions

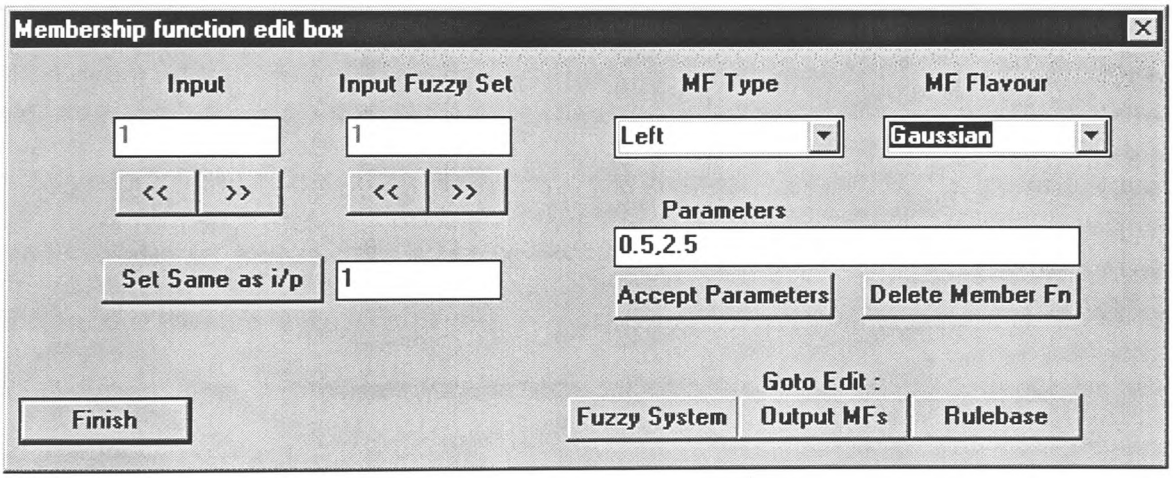

Figure 5.5: Screenshot of input fuzzy set editing window of FTEST

5.3.4 Output sets editing screen.

This interface, shown in figure 5.6, allows the user to enter the parameters for the output sets. In the case of a Zero order system only the constant parameter needs to be set. For a first order or FIR system as many output set parameters as there are inputs need to be specified. The constant parameter is also required for a first order Sugeno system.

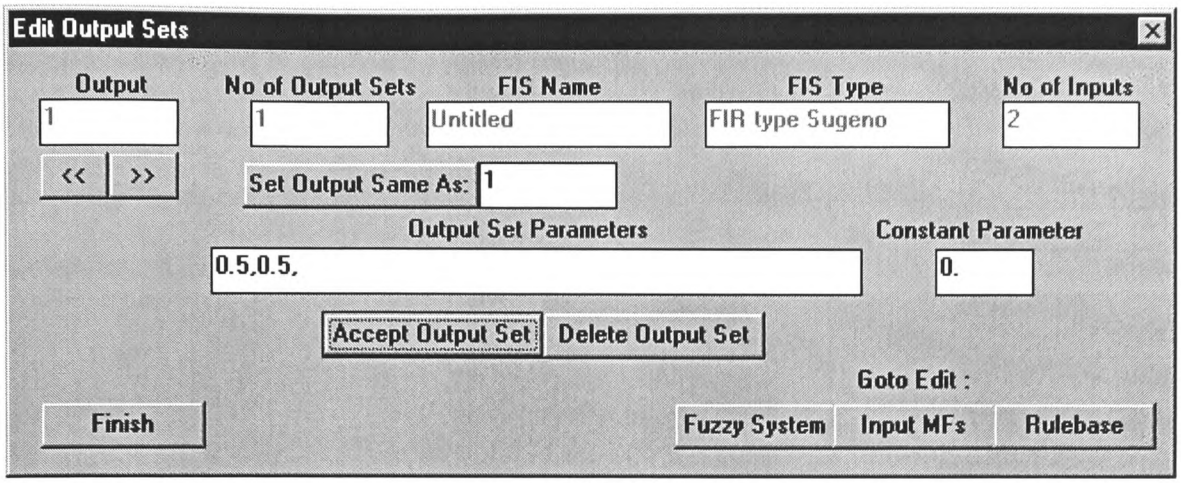

Figure 5.6: Screenshot of Sugeno output set editing window of FTEST

5.3.5 Rules editing screen

This screen, shown in figure 5.7, enables the user to manually enter rules for the fuzzy system.

The rules are of the form:

IF input n is fuzzy set nn AND....AND input m is fuzzy set mm THEN output is output set p.

In the Rule editing box the rules in the above form are coded as:

[n, nn].....[m, mm]: p

Rules can also be read in from a file generated by an external program such as MATLAB provided that they are in the form of pairs of numbers representing the inputs and the fuzzy sets and that every input is associated with an input set. Additionally rules can be generated from the FTEST training dialogue box. The FTEST training dialogue box is discussed in section 5.7.4 following a description of the algorithms used to train the fuzzy systems in this thesis.

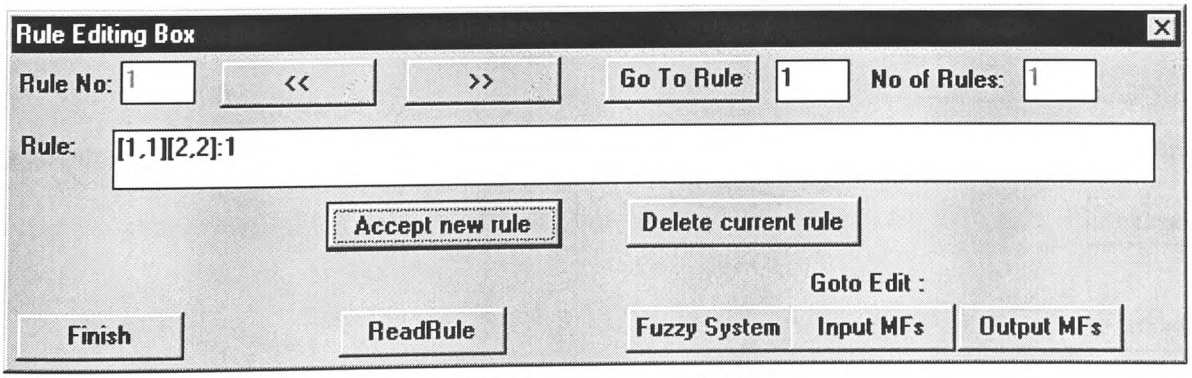

Figure 5.7: Screenshot of Rulebase editing window of FTEST

5.4 Description of Sugeno Inferencing system used as basis for Fuzzy Filters

Each rule consists of N predicate clauses, one predicate clause for each input. Each predicate clause produces a weight which expresses the degree to which that predicate clause is satisfied - a singleton fuzzifier. For Gaussian input sets the weight of the n^{th} predicate clause in the r^{th} rule is given by:

$$Wpc_n^r = Gauss(Ip_n, \mu_f, \sigma_f) \tag{5.3}$$

Where:

$$Gauss(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{\sigma^2}\right] \tag{5.4}$$

The μ_f, and σ_f are the location and width parameters of the membership function describing the fuzzy input set invoked in the predicate clause.

The overall firing weight for the rule is determined by applying a conjunction operator to the vector of weights consisting of all the individual predicate clause weights. The 'min' operator is chosen as the conjunction operator for all the work described here so that the unnormalised firing weight for the r^{th} rule is given by:

$$W_r' = \min\{Wpc_1^r, Wpc_2^r, \ldots\ldots Wpc_N^r\} \tag{5.5}$$

Normalisation is then applied to the firing weight of each rule by dividing by the sum of the firing weights over all the R rules, so that the normalised firing weight for each rule, W_r is given by:

$$W_r = \frac{W_r'}{\sum_{r=1}^{r=R} W_r'} \tag{5.6}$$

Each rule is assumed to have its own output set. As described in section 5.3.3, the FIR type output set consists of an n-tuple of output set parameters (one output parameter for each crisp input). The output for each rule r consists of the scalar product of the N-length vector of crisp input values $\boldsymbol{X}$ with the N-length vector of output set parameters $\boldsymbol{A}_r$, multiplied by the normalised firing weight for that rule as given by equation 5.6. The vector $\boldsymbol{X}$ is:

$$\boldsymbol{X} = \{x_1, x_2 \ldots\ldots x_n\} \tag{5.7}$$

And the output set parameter vector $\boldsymbol{A}_r$ is:

$$\boldsymbol{A}_r = \begin{bmatrix} a_1^r \\ a_2^r \\ a_3^r \\ \vdots \\ \vdots \\ a_N^r \end{bmatrix} \tag{5.8}$$

The final crisp output from the fuzzy inferencing system is obtained by summing the weighted outputs over all the rules. The operation of a first order Sugeno Fuzzy inferencing system can be split into two parts:

- A non-linear mapping of the N-length input vector $\boldsymbol{X}$ into an R-length normalised firing weight vector $\mathbf{W}$. The nature of this mapping is determined by the R rules in the rule base and by the $2.N.F$ input set parameters (referred-to as the 'non-linear parameters').

- For any given normalised firing weight vector, a linear mapping of the input vector X into a single crisp output. This mapping is controlled by the parameters of the output sets (referred- to as the 'linear parameters').

5.5 Training of the linear output set parameters

5.5.1 Overall approach to output set training

The overall approach to training of the linear output set parameters uses input –output data that is representative of the typical inputs to and corresponding desired outputs from the fuzzy inferencing system, called the training data set. For a given training data set, and for given non-linear input set parameters and rules, the output set parameters are chosen so as to minimise the squared error between the actual output of the fuzzy inferencing system for the training data set inputs and the training data set output values. The squared error is evaluated over the whole of the data set. In this way, providing that the training data captures all the important features which might be encountered in the problem domain to which the fuzzy inferencing system is to be applied, the linear output set parameters can be optimised for the given input set parameters and fuzzy system rules.

The application of linear least squares optimisation of output set parameters to first order Sugeno inferencing systems was first used in the hybrid training algorithm of (Jang, 1993). The implementation of the technique described in sections 5.5.2 and 5.5.3 differs slightly from that of (*op cit* Jang 1993), in that a singular value decomposition technique is used rather than an iterative solution to the least squares problem.

5.5.2 Formation of the training problem as a system of linear equations

If there are D sets of N length input data vectors, then the training data can be written as a D by N training data input matrix, $\mathbf{T}$, and a D x 1 vector of training data outputs $\mathbf{O}_t$:

$$\mathbf{T}=\begin{bmatrix} x_1^1\ x_2^1 x_3^1 \ldots\ldots\ldots x_N^1 \\ x_1^2\ x_2^2 x_3^2 \ldots\ldots\ldots x_N^2 \\ : \qquad\qquad\qquad : \\ : \qquad\qquad\qquad : \\ x_1^D\ x_2^D x_3^D \ldots\ldots\ldots x_N^D \end{bmatrix}=\begin{bmatrix} \mathbf{X^1} \\ \mathbf{X^2} \\ \mathbf{X^3} \\ : \\ : \\ \mathbf{X}^N \end{bmatrix} \tag{5.9}$$

For the d^{th} input training data set an R length vector of normalised firing weights, $\mathbf{W^d}$, is computed according to equations 5.5 and 5.6, where:

$$\mathbf{W}^d=\begin{bmatrix} w_1^d\ w_2^d\ w_3^d \ldots . w_R^d \end{bmatrix} \tag{5.10}$$

A D by $N.R$ matrix $\mathbf{B}$ is formed such that:

$$\mathbf{B}=\begin{bmatrix} w_1^1\mathbf{X}^1\ w_2^1\mathbf{X}^1\ w_3^1\mathbf{X}^1 \ldots\ldots\ldots\ldots\ldots w_R^1\mathbf{X}^1 \\ w_1^2\mathbf{X}^2\ w_2^2\mathbf{X}^2\ w_3^2\mathbf{X}^2 \ldots\ldots\ldots\ldots\ldots w_R^2\mathbf{X}^2 \\ : \qquad\qquad\qquad\qquad\qquad\qquad : \\ : \qquad\qquad\qquad\qquad\qquad\qquad : \\ w_1^D\mathbf{X}^D\ w^D\mathbf{X}^D\ w_3^D\mathbf{X}^D \ldots\ldots\ldots\ldots\ldots w_R^D\mathbf{X}^D \end{bmatrix} \tag{5.11}$$

By writing the vector of output set parameters for the rth rule as equation 5.8:

$$\boldsymbol{A}_r=\begin{bmatrix} a_1^r \\ a_2^r \\ a_3^r \\ : \\ : \\ a_N^r \end{bmatrix}$$

Then an $R.N$ length vector $\boldsymbol{P}$ of all the output set parameters can be formed:

$$\boldsymbol{P}=\begin{bmatrix} a_1^1 \\ a_2^1 \\ a_3^1 \\ \vdots \\ a_N^1 \\ a_1^2 \\ \vdots \\ a_N^2 \\ \vdots \\ \vdots \\ a_N^R \end{bmatrix}=\begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \boldsymbol{A}_3 \\ \vdots \\ \vdots \\ \boldsymbol{A}_R \end{bmatrix} \qquad 5.10$$

The output of the fuzzy inferencing system for each of the D sets of training data input vectors will be given by the elements of the D by 1 output vector $\boldsymbol{O}$ given by:

$$\boldsymbol{O} = \boldsymbol{B}.\boldsymbol{P} \qquad 5.11$$

This represents a set of linear equations. In order to force the fuzzy system to give the correct output for the input training data it is necessary to solve equation 5.11 for $\boldsymbol{P}$ when $\boldsymbol{O} = \boldsymbol{O}_t$. For real data sets in general $D \neq R..N$ and therefore equation 5.11 cannot be solved uniquely for $\boldsymbol{P}$. Even if it is arranged for $D = R.N$, the matrix $\boldsymbol{B}$ is likely to be singular for real data or very nearly singular (ill-conditioned). Thus for real data an exact unique solution to equation 5.11 cannot be obtained. However provided it is arranged that $D > R.N$ (i.e. sufficient training data is used) then equation 5.11 represents an overdetermined set of linear equations. A least squared error solution can then be obtained for $\boldsymbol{P}$ (Lawson and Hanson, 1974). One solution to such an overdetermined set of equations is solved using the pseudo inverse of $\boldsymbol{B}$, labelled $\boldsymbol{B}^{\Diamond}$:

$$P=B^{\Diamond}O_t \tag{5.12}$$

where the pseudo inverse, $B^{\Diamond}$, is given by :

$$B^{\Diamond}=(B^T B)^{-1}B^T \tag{5.13}$$

However, the matrix $B^T B$ is often also ill conditioned for real data, making evaluation of its inverse unreliable. In order to avoid this problem a least squared error technique for solving equation 5.11 for the overdetermined case using a singular value decomposition of B is used. This is described in section 5.5.3

5.5.3 Least squares solution by Singular Value Decomposition

It can be shown from linear algebra, e.g. (*op cit* Lawson and Hanson, 1974), that any I by J matrix **B** having I greater than or equal to J can be factored into three matrices U, W, and V such that:

$$B=U.W.V^T \tag{5.14}$$

Where U is I by J, W is J by J, and V is J by J. The matrix U has columns which are orthonormal and therefore is an orthogonal matrix. V is also an orthogonal matrix. The matrix W has values on the main diagonal only, which are also all positive, and all the off-diagonal values of W are zero. The elements on the diagonal of W are called the 'Singular Values'. Any matrix can be decomposed as equation 5.14 even if it is ill-conditioned or singular. Singular matrices have values of zero for some of the singular values. If a matrix is ill-conditioned then the singular values will be small.

For a non-singular matrix $\boldsymbol{B}$ then from equation 5.14 :

$$\boldsymbol{B}^{-1} = \boldsymbol{V}.\boldsymbol{W}'.\boldsymbol{U}^{T} \qquad 5.15$$

where $\mathbf{W}'$ is the matrix $\mathbf{W}$ with the diagonal elements (the singular values), w_{ij} replaced by $1/w_{ij}$ $_{(i=j)}$. For the case of a singular matrix $\boldsymbol{B}$ the diagonal elements of $\boldsymbol{W}'$ will be infinite, and for an ill-conditioned $\boldsymbol{B}$ the elements will be very large. If the zero or small valued elements of $\boldsymbol{W}$ are replaced by zero instead of $1/w_{ij}$ to form the matrix $\mathbf{W}^{*}$ then the solution to with $\mathbf{O} = \mathbf{O}_t$ given by:

$$\boldsymbol{P} = \boldsymbol{V}.\boldsymbol{W}^{*}.\boldsymbol{U}^{T} \qquad 5.16$$

gives a solution for the parameter vector $\boldsymbol{P}$ which is the least sum of squared error (Press *et al.*., 1994). i.e. equation 5.16 gives the vector $\mathbf{P}$ such that

$$\left|\boldsymbol{B}.\boldsymbol{P} - \boldsymbol{O}_t\right|$$

is a minimum.

5.5.4 Singular Value decomposition applied to training the linear output set parameters

In general the equation 5.11 will consist of an overdetermined set of equations for real training data. In fact it is desirable to ensure that the number of training data sets comfortably exceeds the number of linear parameters to be adjusted. The result of equation 5.16 also applies to the overdetermined case. In order to use Singular Value Decomposition to train the output set parameters for a given set of non-linear parameters, the matrix $\boldsymbol{B}$ is computed over the training data set. The Singular Value Decomposition of $\boldsymbol{B}$ is then carried out and the diagonal elements

of W (the singular values) are then examined. If any are less than a number ε the corresponding elements of W'' are set to zero. For elements greater than ε the corresponding elements in W'' are set to their reciprocal. The value of ε is dependent on the precision of the floating point representation used in the software and is set to 1×10^{-6} in the program FTEST. The vector of all the output sets parameters, P is then calculated using equation 5.16.

5.6 Implementation of output set training in the FTEST application software

This section uses an example to describe the implementation of linear output set training in the FTEST application software.

Example:

The example used to illustrate the method of training the linear output sets described in section 5.5 is a fuzzy inferencing system which approximates a simple function of two input variables, x_1 and x_2 over a universe of discourse from 0 to 10. The function is:

$$y = x_1^2 + x_2^2 \qquad 5.17$$

The function defined by equation 5.17 and the fuzzy approximator to it are referred to as 'SQR2' for brevity in the following text. The initial (untrained) fuzzy inferencing system used to approximate this function was configured to have three fuzzy sets per input, giving a maximum of $3^2 = 9$ rules. All nine of the possible rules were used and in the notation described in section 5.3.4 these were as shown in table 5.1:

[1,1][2,1]:1	[1,2][2,1]:4	[1,3][2,1]:7
[1,1][2,2]:2	[1,2][2,2]:5	[1,3][2,2]:8
[1,1][2,3]:3	[1,2][2,3]:6	[1,3][2,3]:9

Table 5.1 Rulebase for Fuzzy system approximating SQR2

The output sets, which were to be trained were initially all set, using the notation of section 5.3.2 to the parameter vector [5,5]. The Gaussian input sets were spread evenly about the universe of discourse as shown in table 5.2. These input sets were not trained in this example

	Input 1	**Input 2**
Fuzzy set 1	$\mu = 2, \sigma = 3$	$\mu = 2, \sigma = 3$
Fuzzy set 2	$\mu = 5, \sigma = 3$	$\mu = 5, \sigma = 3$
Fuzzy set 3	$\mu = 8, \sigma = 3$	$\mu = 8, \sigma = 3$

Table 5.2: Parameters for Input fuzzy sets for function SQR2

Since there were eighteen linear parameters to adjust, the training data set needed to contain at least eighteen input–output data pairs. In this case a set of a hundred input-output data pairs were used, resulting in an overdetermined system corresponding to equation 5.11. These training data pairs were created in MATLAB saved as a matrix of floating point numbers, and read in by the FTEST application software. The FTEST training window also allows the calculation of mean squared error and mean absolute deviation before and after training. For the example here the mean squared error was 859.1 over the training data set before training of the output sets, and 24.5 after training using least squares. It should be noted that optimising the total squared error will also optimise the mean squared error. However optimising the mean squared error does not optimise the total squared error in general. After training, the output sets were as shown in table 5.3.

Output set r	Parameter 1 (a_1^r)	Parameter 2 (a_2^r)
1	-3.33	-4.04
2	-0.125	3.44
3	-3.02	14.3
4	3.62	1.69
5	4.61	6.81
6	2.93	10
7	14.3	-6.26
8	12.3	3.58
9	12.3	16.3

Table 5.3: Output set parameters for output fuzzy sets for function SQR2 after training

The adjustment of the output set parameters resulted in a large improvement in the ability of the fuzzy system to approximate the function SQR2. Clearly, though, the error is still quite large and the fuzzy inferencing system is not optimal, as the input set parameters have not been adjusted. The output sets as defined in table 5.3 are only those that produce least squared error given the input set parameters of table 5.2. Inspection of table 5.3 also shows very little discernible pattern in the values of the output set parameters. It would be difficult to predict these parameters from an intuitive linguistic description of the problem. This is partly because the problem being tackled is a mathematical function that is not easily described in natural language, and partly because the Sugeno inferencing method uses a mathematical description for its output sets which cannot be easily identified with a linguistic label. Thus this example gives rise to two points:

> There is a class of problems, of which the approximation of mathematically described functions is an example, for which adequate natural language descriptions do not exist. For problems of this type the ability of fuzzy systems to capture natural language or intuitive descriptions is of no advantage. If a fuzzy inferencing system is to be applied to these problems, therefore, it must be capable of being trained using numerical exemplar data.

Fuzzy inferencing systems that use the Sugeno inferencing method lose some of the links to intuitive natural language reasoning. In such fuzzy systems training of the output set parameters is likely to be necessary. For first order Sugeno systems the least squares method described above is an effective one-pass technique.

5.7 Training of the non-linear input set parameters.

5.7.1 Overview of problem

In section 5.6 the least squares approach was applied to train the output set parameters, but it was pointed out that this technique could not be used to train the non-linear input set parameters. The problem can be stated as:

> Adjust the parameters so as to minimise a measure of the error between the training data output and the actual output obtained from the fuzzy inferencing system when it is presented with the training data inputs. The error measure should be taken over a set of input-output training data so as to ensure adequate generalisation from the trained system.

If there are P parameters to be adjusted, the problem can be visualised as a P+1 dimensional landscape or 'error terrain'. P of the dimensions represent the values of the input set parameters and the $P+1^{th}$ dimension or height dimension represents the error measure over the training data set. The problem is then to find the P co-ordinates of the 'lowest' point in this terrain. In general the error terrain will have many local minima and finding the global minimum may well be impossible at least within a reasonable time of computation. The problem statement can therefore be re-stated as that of finding a 'good local minimum' within a reasonable time of computation. An acceptable level of error must be chosen beforehand in order to define a 'good local minimum'. It is an advantage for training techniques used to optimise fuzzy inferencing systems to allow for the existence of local minima. Those

techniques which do not allow for local minima can become trapped in a local minimum which does not satisfy the criterion of a good local minimum. Pure gradient descent techniques such as the back-propagation technique used in the ANFIS algorithm (*op cit* Jang, 1993) do not allow for local minima. Pure gradient descent is appropriate for those problems in which the topology of the error terrain is roughly known so that any local minima can be avoided. Gradient descent can also be used when the fuzzy inferencing system can incorporate enough expert knowledge before training that the system can be assumed to be close to a good local minimum. When neither of these conditions obtain then more robust search techniques are needed.

5.7.2 Simulated Annealing

Simulated annealing is a stochastic search technique which can be used to find minima in an error terrain. It possesses the advantage that it can avoid becoming trapped in local minima. The origins of the algorithm lie in work carried out by (*op cit.* Metropolis *et al.*, 1953) to simulate thermodynamic systems using Monte Carlo techniques. In such thermodynamic systems in equilibrium at a temperature T the probability distribution of the energy of the system is given by the Boltzmann distribution (Mandl, 1973):

$$P(r)=\frac{1}{Z}\exp\left[-\frac{E_r}{KT}\right] \qquad 5.18$$

where r is an index for a particular discrete microstate of the system, and E_r is the energy of that microstate (different microstates can have the same energy). K is the Boltzmann constant. The quantity Z is the partition function for the system. It acts as a normalising constant and represents the sum over all microstates of the system:

$$Z = \sum_r \exp\left[-\frac{E_r}{KT}\right] \qquad 5.19$$

The Metropolis algorithm simulates a system with the probability distribution given by equation 5.18 in the following way:

The system is initialised to some state r_1 and the energy $E(r_1)$ is calculated. A random change is made to the system to some other state r_2 with energy $E(r_2)$ and the change in energy ΔE is calculated.

$$\Delta E = E(r_1) - E(r_2) \qquad 5.20$$

If the change in energy ΔE is negative then the random change is accepted. If ΔE is positive then the change is accepted with probability given by:

$$P = \exp\left[-\frac{\Delta E}{KT}\right] \qquad 5.21$$

In a physical system if the temperature of the system is slowly reduced so that the system remains in equilibrium throughout, then the system will relax to a minimum energy state, avoiding intermediate metastable states. This process is called annealing and results in well-ordered systems with low energy. In the Metropolis algorithm the annealing process is achieved by reducing the temperature T after a sufficient number of random changes have been made at a fixed temperature. The process governing the reduction in temperature is called the 'Annealing Schedule'.

(Kirkpatrick *et al.*, 1983) have applied the Metropolis algorithm to n-dimensional optimisation problems. The energy is replaced by an objective function that depends on the parameters whose optimal values or configuration is sought. This objective function is chosen to reflect the goodness of the parameter configuration. For example in a function fitting problem, the mean squared error over the domain of the function between the actual data to be fitted and the proposed function could be used as an objective function. A potential advantage of applying the Metropolis algorithm to optimisation problems is its ability to avoid local minima in the error terrain which correspond to the metastable energy states of a physical system. It achieves this by having a high probability of accepting a random change which results in a higher system energy state when the temperature is high. In this way the system has a good probability of stepping out of small local minima during the early high temperature phase of the annealing process. During the final low temperature phase of the annealing schedule the algorithm performs a random descent in which only downward steps are accepted

Although the simulated annealing appears relatively easy to implement, the choice of annealing schedule and initial temperature can have a large effect on the final result and are problem dependent. This has the effect that some insight into the nature of the problem is required to make effective choices. A typical tactic for determining effective starting temperatures is to perform short initial runs of the algorithm at various temperatures and to observe the change in the error measure. Such an initialising process is difficult to automate and therefore a drawback to the algorithm. It is also important that the method of making the random system state changes allows adequate coverage of the parameter space to be searched. This also requires some insight into the problem being solved. Nevertheless the algorithm has been used to solve problems whose computational complexity scales at a rate that is greater than can be expressed as a polynomial function of the problem size (NP complete problems). Its main competitor for such search problems is the class of Genetic Algorithms (Goldberg, 1989). The

Tabu search technique has also been used for combinatorial optimisation and is compared with simulated annealing in (Chiang *et al.*, 1992).

The Metropolis algorithm has been successfully applied to combinatorial optimisation problems such as the 'travelling salesman' problem (Kirkpatrick *et al.*, 1983), (Kirkpatrick, 1984). In such problems the states of the systems are represented by combinations of discrete parameters. One of the chief justifications for applying the algorithm to optimisation problems other than those statistical mechanics problems for which it was originally conceived is the similarity of such systems to each other. However, the original problem domain governed by the Boltzmann distribution and for which the algorithm is a sampler, has discrete variables and more closely represents a combinatorial optimisation rather than a continuous parameter optimisation.

The application of the algorithm to optimisation over continuous variables is discussed by (Vanderbilt and Louie, 1984). Optimisation over continuous variables raises further issues over and above those raised for combinatorial optimisation. These issues are the choice of the direction and size of the random perturbations made to the continuous parameters and whether such changes should vary with the annealing temperature. Small steps, particularly in the early stages of the annealing schedule are inefficient, whereas very large steps result in too high a rejection rate. It is suggested by (*op cit* Vanderbilt and Louie, 1984) that optimum efficiency occurs when half the random changes are rejected. They also propose and demonstrate a sophisticated method of controlling the size and direction of the random steps in the n-dimensional parameter space. Their scheme is based on deriving a measure of the local error space topography from the random walk segment of accepted solutions of the last training epoch. This measure is used to choose the values of a covariance matrix which controls the distribution of the random steps. A growth factor ensures that the covariance matrix is set so

that a free random walk would cover ever-increasing volumes of the parameter space. In the constrained error space the covariance matrix grows to fit the shape of the error space so that the step size only grows in the direction of downhill moves. The annealing schedule used is a geometric series with a common ratio χ between zero and unity so that the temperature at the n^{th} iteration is $\chi^n T$.

5.7.3 Simulated annealing applied to optimisation of fuzzy inferencing systems

The training of a fuzzy system involves optimising a multidimensional parameter set, which is the type of task for which simulated annealing is ideally suited. Nevertheless, apart from a recent paper by (Garibaldi and Ifeachor, 1999), in which simulated annealing is applied to the optimisation of a fuzzy expert system of the Mamdani type, it is not believed that simulated annealing has appeared in the literature applied to fuzzy system training. As part of the work described in this thesis, the simulated annealing algorithm was applied in six different ways to the problem of training fuzzy systems from training data. These six methods were:

1. Simulated annealing applied to search for input and output parameters
2. Simulated annealing applied to search for input parameters only
3. Simulated annealing applied to search for output parameters only
4. Simulated annealing applied to search for input parameters with linear least squares applied to optimise the output set parameters for the candidate input parameters before assessing the error.
5. Simulated annealing combined with downhill simplex algorithm to search for input parameters with linear least squares used to optimise output set parameters
6. Simulated annealing used with an automatic rule-generating algorithm to select the optimal set of rules. The input set parameters were held constant for this method, but the output set parameters were optimised using linear least squares for each candidate set of rules.

5.7.4 Simulated annealing used for training input and output parameters

This section covers the first three methods enumerated in section 5.7.3 since the techniques used are essentially the same for all three methods. The differences between methods are solely the choice of parameters to be adjusted by the algorithm. The basic steps of the algorithm are shown in the flow diagram figure 5.9.

The following list describes some of the details of the implementation of the first three methods in the application software FTEST.

1. The initial value of the temperature parameter is input by the user through the training dialog box in FTEST. The training dialog box is shown in figure 5.8. The initial temperature needs to be large compared to the largest changes in error which are to be found in the error space. A feel for the magnitude of the error changes that are likely to be encountered can be obtained by carrying out a few short trial training runs.

2. The number of iterations per epoch can be set in the dialog box as can the maximum number of epochs to be run. A minimum error can also be entered, which, if achieved during training, automatically terminates the training run.

3. The initial input and output parameters of the fuzzy inferencing system are normally manually entered through the Fuzzy system editing dialog boxes. If desired however, the 'initialise system' button will automatically generate a default set of input sets, output sets, and rules. The generated input sets are evenly distributed over the input universe of discourse set in the 'fuzzy system' editing dialog box. The output set parameters are all set to a default value of one. This automatic rule and parameter generating option was written-in to the software to implement the rule training option (method 6). In general the

parameters and rules should be manually entered to initialise the fuzzy system to a good initial configuration, taking full advantage of 'expert knowledge'.

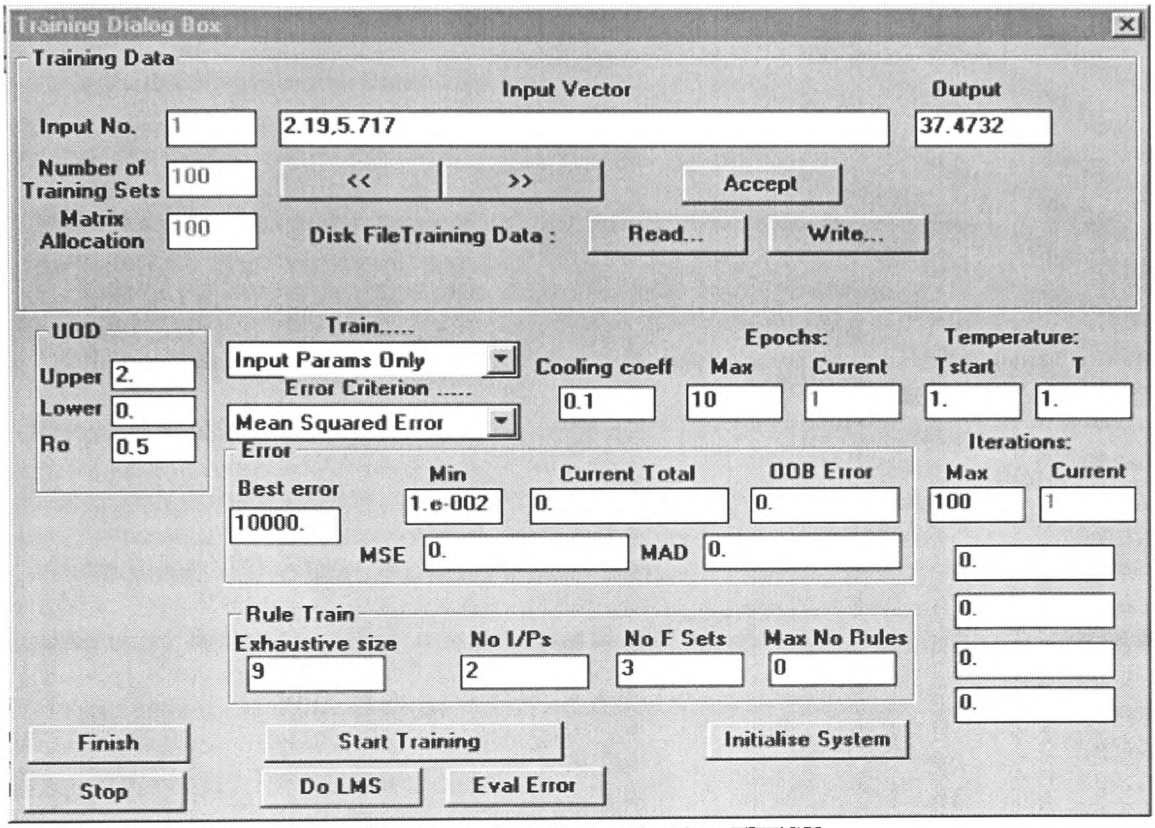

Figure 5.8: Screenshot of training dialog box in application FTEST

4. For training of the input parameters only, random perturbation of the parameters are made as follows:

   One of the inputs to the fuzzy system is chosen at random with even probability of choosing any of the inputs. One of the possible fuzzy sets covering this input's space is then randomly chosen with all sets being equiprobable. The type of fuzzy set is then determined (Gaussian or Trapezoidal) and hence the number of parameters defining it. For a Gaussian set one of the two parameters is chosen randomly. The width parameter is allowed to change to between 10% and 150% of its previous value. The position parameter is allowed to change by up to ± 50% of its previous value. For trapezoidal input sets the four parameters defining the trapezoid are also randomly varied such that

the effective centre and width of the fuzzy set can evolve from their existing values. The four parameters of trapezoidal input sets are constrained by the software to maintain their ordering. Thus for the example shown in figure 5.4 the parameters c, d, e, f are constrained so that $c \leq d \leq e \leq f$.

5. When the output set parameters are being trained random changes are made as follows:

   One of the output sets is chosen at random with equal probability for each set. Within that output set of parameters one of the parameters is chosen at random. The selected parameter is randomly perturbed by up to ± 50% of its existing value.

6. If both inputs and outputs are being trained at once a random 'coin toss' is made at each iteration to determine whether an input or output set is to be perturbed. Thereafter the random moves are either as those described above for perturbations of the inputs or outputs.

7. The basic error measure used as a cost functional is the mean squared error over the training data set. If the fuzzy system produces an output O_t for the t^{th} training data input and the actual training data output for that input is OT_t then the mean square error E_1 over the set of T training inputs and outputs is:

$$E_1 = \frac{1}{T}\sum_{t=1}^{t=T}(O_t - OT_t)^2 \qquad 5.22$$

An additional error measure is added to E_1 to derive an overall cost functional. This error is called the Out Of Bounds error (OOB error) and is included to increase the efficiency of the training algorithm. The Out of Bounds error is computed from the input parameters. During training the input parameters may be perturbed such that the area of coverage of one of the fuzzy sets falls outside the universe of discourse of the fuzzy system. In general, the training

data does not fall outside the universe of discourse. Therefore any fuzzy set whose area of coverage is far outside the universe of discourse will never be activated by the training data and will have no effect on the output and hence the error. Any such fuzzy set is therefore effectively removed from the fuzzy system. A further random perturbation of this set causes no change in the error unless it is brought back into the universe of discourse. The out of bounds error measure penalises the removal of fuzzy sets to a distance defined by a radius parameter beyond the edges of the universe of discourse. The radius parameter, R_0 , is set before training in the training dialog window. For the n^{th} input and the f^{th} Gaussian input fuzzy set, *Gauss(centre, width)* , with the upper and lower bounds of the universe of discourse denoted by *UOD_lo* and *UOD _hi,* the out of bounds cost is given by:

$$OOBCost(n,f)=\exp\left[\frac{(UOD\_lo-centre+3.width)}{R_0}\right] \qquad centre+3.width<UOD\_lo$$

$$OOBCost(n,f)=0 \qquad UOD\_lo<centre<UOD\_hi$$

$$OOBCost(n,f)=\exp\left[\frac{(centre-3.width-UOD\_hi)}{R_0}\right] \qquad centre-3.width>UOD\_hi \qquad 5.23$$

The out of bounds measure E_2 is computed over all inputs and all fuzzy sets and is given by:

$$E_2=\max_{N,F}(OOBCost(n,f)) \qquad 5.24$$

Where $\max_{N,F}(.)$ denotes the maximum over N input sets and F input fuzzy sets.

For the n^{th} input and the f^{th} trapezoidal input fuzzy sets, *Trapz(a ,b ,c ,d)* with parameters *a, b, c, d,* the out of bounds cost is given by:

$$OOBCost(n,f)=MAX[OOBCost\_lo(n,f),OOBCost\_hi(n,f)] \qquad 5.25$$

Where:

$$
\begin{aligned}
&OOBCost\_lo(n,f) = \exp\left[\frac{(UOD\_lo - a)}{R_0}\right] && a < UOD\_lo \\
&OOBCost\_lo(n,f) = 0 && UOD\_lo < a \\
&OOBCost\_hi(n,f) = \exp\left[\frac{d - UOD\_hi}{R_0}\right] && d > UOD\_hi \\
&OOBCost\_hi(n,f) = 0 && UOD\_hi > d
\end{aligned}
\qquad 5.26
$$

The overall error is given by $E_t = E_1 + E_2$. If the change in overall error from the previous system state, ΔE_t , is positive (i.e. the error is worse than the previous error) then the function *metrop* is evaluated:

$$metrop = \exp\left[-\frac{\Delta E_t}{T}\right] \qquad 5.27$$

If the value of *metrop* is greater than a uniformly distributed pseudo random number then the change in the system state is accepted.

8. The Annealing schedule sets the temperature, $T[e+1]$ at epoch $e+1$ as:

$$T[e+1] = \frac{T[0]}{1 + \alpha . T[e]} \qquad 5.28$$

Where α is the cooling coefficient, which determines the rate at which the temperature is reduced during training, and $T[0]$ is the initial temperature at the start of the first training epoch.

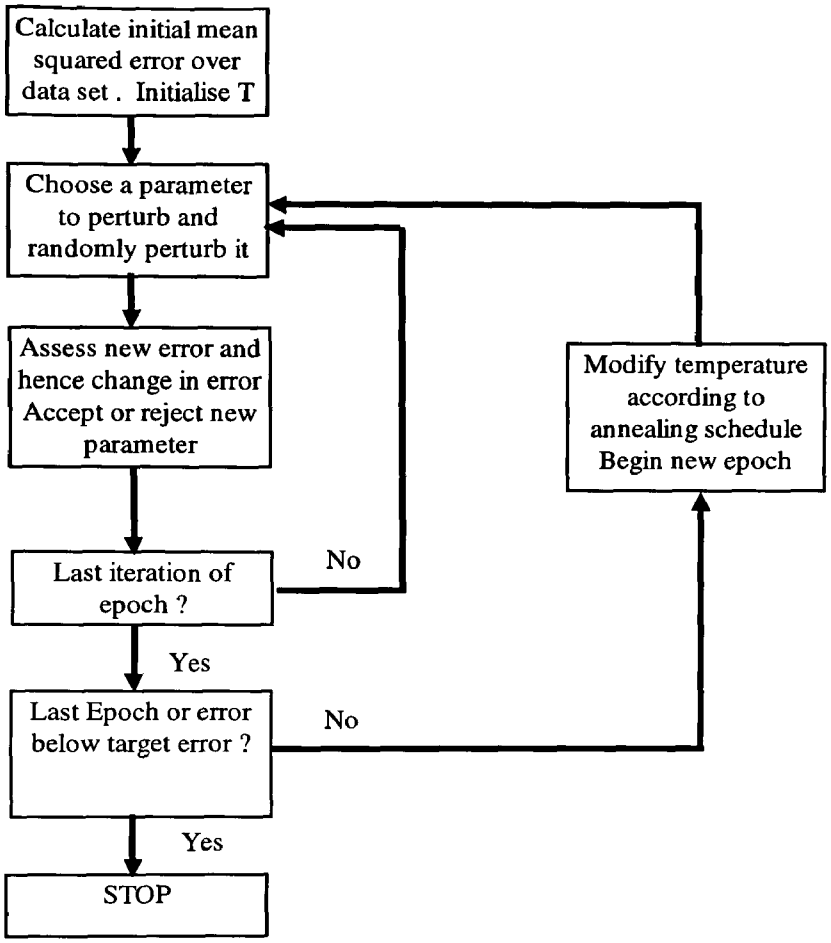

Figure 5.9: Flow diagram for training of input and output parameters using simple simulated annealing

5.8 Simulated annealing used to train fuzzy inferencing system to approximate SQR2

The following examples demonstrate the three training modes described above applied to the training of a fuzzy inferencing system to approximate the function 'SQR2'

5.8.1 Example: Training of output set parameters only

The first example used simulated annealing to train the output set parameters only, as described in section 5.7.4. The starting point for the fuzzy system was identical to the fuzzy system trained using linear least squares in order to allow some comparison. After training for 1500 epochs of 120 iterations each, with an initial temperature of 10 the mean squared error (MSE) was reduced from the initial value of 859.1 to a value of 8.5. The mean absolute deviation

(MAD) was also monitored, but was not a part of the objective function. The MAD reduced from 22.5 to 2.3. Figure 5.10 shows the change in error with respect to the epoch number during training.

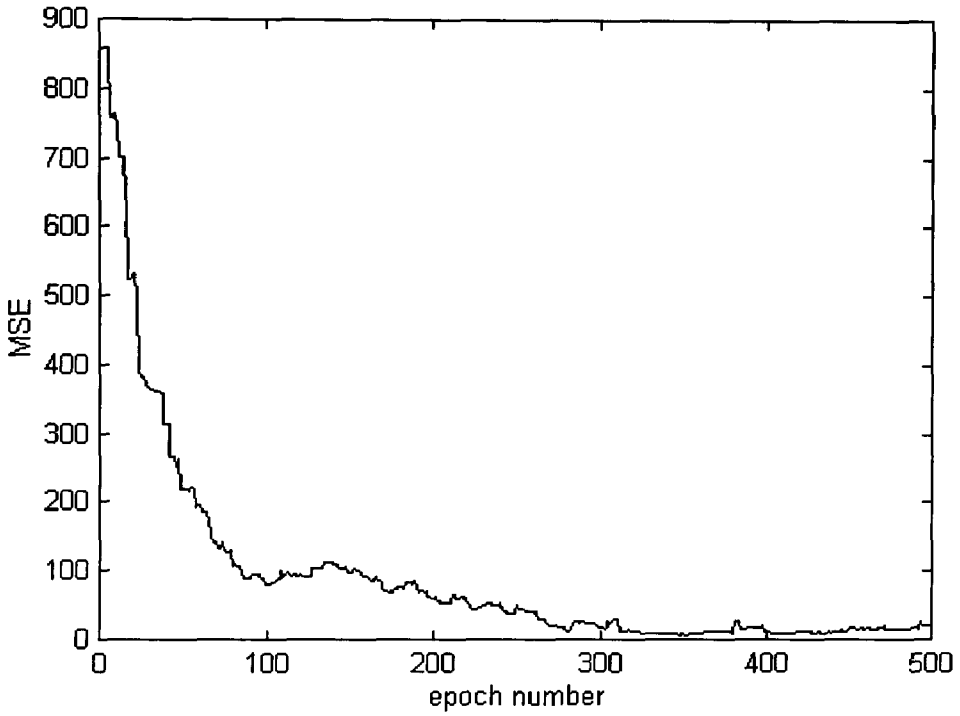

Figure 5.10: Plot of error versus epoch number for simulated annealing training of output parameters

The MSE compares well with the result of a MSE of 25 obtained using the linear least squares method (section 5.6). This illustrates the fact that optimising the squared error does not automatically optimise the mean squared error. Using MSE as an error measure a few large errors may be tolerated in a large data set if allowing them improves the mean squared error calculated over all the data. By contrast such large errors can dominate a squared error criterion.

The least squares method of training the output sets has an advantage when comparing speed of execution however, since the simulated annealing algorithm took four orders of magnitude longer to run in this case than the one-pass least squares method. Moreover, in training fuzzy

systems the ability to generalise from the training data to another set of test data and hence to real data is an important criterion of successful training. In this case the fuzzy inferencing system trained using simulated annealing gave a MSE of 8.8 for a further test data set, whilst the linear least squares-trained system gave a MSE of 10 on this test data. In this case at least, then, the difference between fuzzy systems using SA to train the output sets and those trained using linear least squares was not so clear on further test data. However it could be argued that the linear least squares solution showed greater variability between data sets and therefore did not generalise as well.

The output sets after training using simulated annealing are tabulated in table 5.4.

Output set r	Parameter 1 (a_1^r)	Parameter 2 (a_2^r)
1	0	8
2	0	0
3	0	16.4
4	0	0
5	0	0
6	13	7.2
7	13.7	0
8	15.3	0
9	8.7	17.1

Table 5.4: Output set parameters for output fuzzy sets for function SQR2 after training using simulated annealing.

It might be conjectured that those output set parameters which have values of all zeros correspond to rules which could be removed from the fuzzy system rule base and have no effect on the output of the system. However, the performance of the fuzzy approximator after training of only the output sets leaves some room for improvement. Training of the input sets offers the possibility of improving its performance, and therefore removal of rules before investigating the effects of training the input sets would be premature.

Application of the simulated annealing routine to the output sets after they had been initially trained using the least squares method improved the MSE from 25 to 3.4. In this application of the simulated annealing routine the temperature was set to a very low value and 1000 iterations were used. This results in the algorithm becoming a simple random search around the starting point with greedy descent. The error on the test data was 3.8 after this further training. This improvement again illustrates the difference between a least squares error criterion and a minimum mean squared error criterion.

5.8.2 Example: Training of input sets only

In this example the simulated annealing algorithm was used to train the input sets only, as described in section 5.7.4. The output sets were initialised to the values given by the least squares algorithm, so that the initial MSE was 25. Thereafter during the training the output sets were left unchanged. After training the mean squared error reduced from 25 to 3.8. The error on the test data set was 3.9. The input set parameters after training are shown in table 5.5

	Input 1	**Input 2**
Fuzzy set 1	$\mu = 2.63$	$\mu = 2.1$
	$\sigma = 2.95$	$\sigma = 2.7$
Fuzzy set 2	$\mu = 6.7$	$\mu = 5.66$
	$\sigma = 2.78$	$\sigma = 3.17$
Fuzzy set 3	$\mu = 8.1$	$\mu = 1.025$
	$\sigma = 3.2$	$\sigma = 0.1$

Table 5.5: Parameters for Input fuzzy sets for function SQR2 after training using simulated annealing

The success of this algorithm in the training of the input sets is dependent on the settings chosen for the initial temperature and the cooling coefficient. A high initial temperature in conjunction with too high a rate of cooling can cause the algorithm to become trapped in a local minimum. It would be possible to detect this condition and abort the training run by monitoring the change in error over several epochs. In practice in the FTEST application software this process is carried out manually by periodically checking the progress of the algorithm. Where the error is observed to settle quite early at a high value, and particularly where a much lower 'best error' has been observed then the training process is likely to have become trapped in a local error minimum (the best error is monitored in the training dialog box). In this case manual intervention is required to reset the cooling rate and possibly the initial temperature. This requirement for intervention is a drawback for this version of fuzzy system training using simulated annealing and is more noticeable for input parameter training. It is conjectured that the need for intervention is greater for input set parameter training because the space defined by the error as a function of the input parameters contains more local minima than the error space for the output parameters alone. One of the major contributors to this sensitivity to cooling coefficient is that the algorithm is not efficient in searching out the local minimum in its vicinity. Thus during the early high temperature phase of training areas of parameter space with potentially deep error minima are abandoned too soon by the search without finding the true local minimum. Later in the low temperature phase of the search the algorithm cannot re-visit these minima, as the temperature is then too low to allow the search to move from its existing local minimum. Figure 5.11 shows the change in error as a function of the epoch number for this example.

5.8.3 Example: Training of input and output sets using simulated annealing

The third example used a combination of the routines used in the examples of sections 5.8.1 and 5.8.2. The routines were combined as described in section 5.7.4. The initial configuration

of the fuzzy system was the same as that used to test the least squares method. After training for 2000 epochs of 100 iterations each, starting with an initial temperature of 4 and a cooling coefficient of 0.05, a final mean squared error of 10 was achieved. The MSE over the verification test data was 12.4. The plot of error Vs epoch number is shown in figure 5.12 and the trained input and output set parameters are shown in tables 5.6 and 5.7.

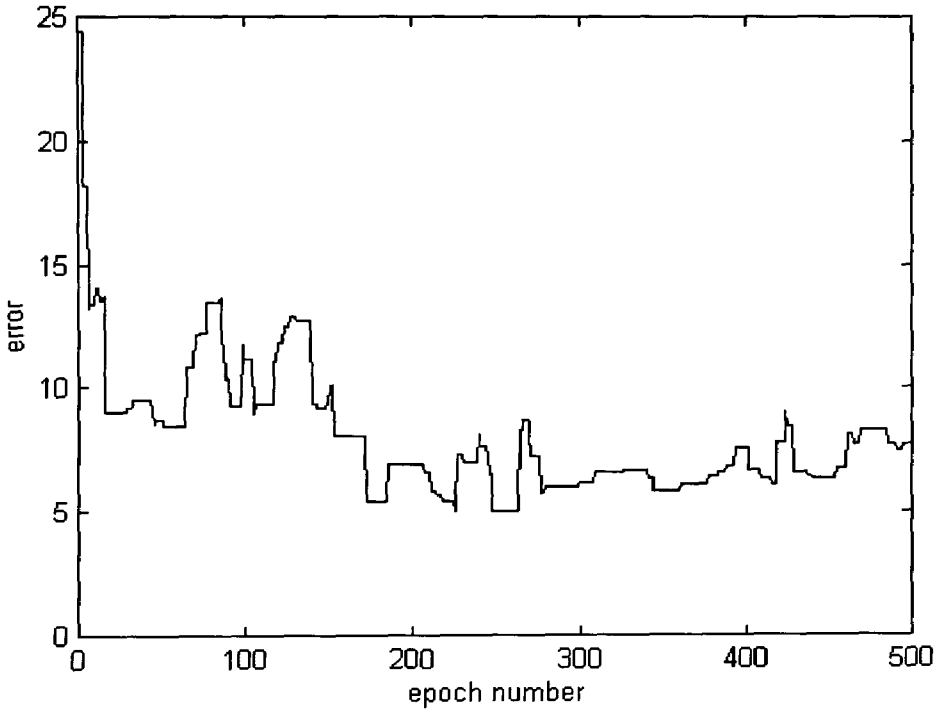

Figure 5.11: Plot of epoch error versus epoch number for example of section 5.8.2

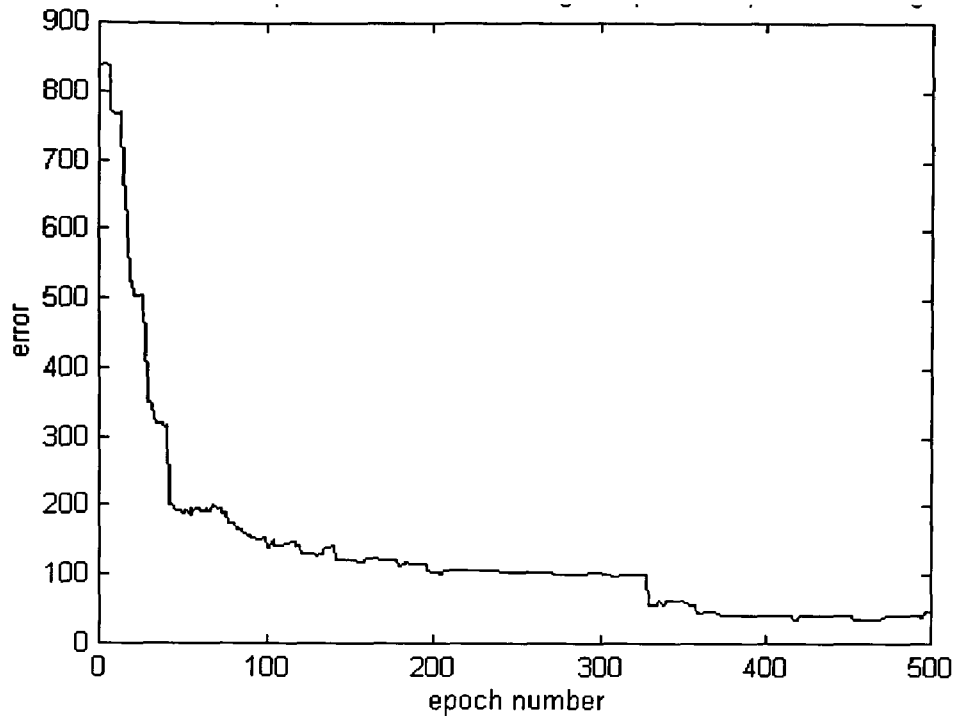

Figure 5.12: Plot of error versus epoch number for example of section 5.8.3

Output set r	Parameter 1 (a_1^r)	Parameter 2 (a_2^r)
1	0	0
2	0	0
3	4.1	8.9
4	10.9	0
5	0	0
6	0	14.4
7	7.67	0
8	0	0
9	11	8.39

Table 5.6: Output set parameters for function SQR2 after training for example of section 5.8.3.

	Input 1	Input 2
Fuzzy set 1	$\mu = 6.1\ \sigma = 0$	$\mu = 2.9\ \sigma = 2.6$
Fuzzy set 2	$\mu = 6.5\ \sigma = 1.5$	$\mu = 5\ \sigma = 0$
Fuzzy set 3	$\mu = 7.5\ \sigma = 0$	$\mu = 0.2\ \ \sigma = 0$

Table 5.7: Input set parameters for function SQR2 after training for example of section 5.8.3.

The disappointing performance of this algorithm using simulated annealing for both inputs and outputs is fairly typical of this training method. This is due to the inefficiency with which the algorithm selects candidate moves. The handicapping effect of this inefficiency becomes more pronounced the larger the search space becomes. In this case the search space is larger because of the dimensionality of the problem and thus the search space has been increased from 12 for the input sets only and a dimensionality of 18 for the output sets only, to a combined dimensionality of 30. Since the search space volume scales the power of the problem dimensionality the efficiency of the search algorithm becomes acutely important. Assuming that all the parameters to be searched have an equal and bounded range of possible values R_p then the volume to be searched for a P parameter problem is given by:

$$VOL_{SEARCH} \propto {R_P}^P \tag{5.29}$$

The potential difficulties imposed by this increase in the problem search-space become even more apparent when the scaling of fuzzy system dimensionality with number of inputs is discussed in section 5.13. Because of the importance of the efficiency of the training methods in searching parameter space the methods described in the following section were adopted to improve the searching efficiency of the algorithm.

5.9 Simulated annealing combined with least squares to train input and output sets

5.9.1 Description of training method

The training method described in this section combines the least squares approach for training the output parameters with simulated annealing to train the input parameters (method 4 of section 5.7.3). The algorithm is similar to the hybrid method described in (*op cit* Jang, 1993), but whereas in that work backpropagation is used to optimise the input parameters, simulated annealing is substituted in the method described here. The overall method is shown in the flow diagram of figure 5.13.

The motivation for the adopting this approach is to try and reduce the effective dimensionality of the search space for the simulated annealing algorithm by taking advantage of the comparative computational efficiency of the least squares approach for improving the output set parameters.

5.9.2 Example: Training of input and output sets using simulated annealing and least squares.

In order to compare this approach with the other approaches described in this chapter the example of training a fuzzy system to approximate the function SQR2 is again used. In this example after a few trial runs the most effective training parameters were found to be: initial temperature = 1, cooling coefficient = 0.4, number of iterations per epoch = 100, number of epochs=100. The initial MSE was 25 and the final MSE on the training data was 0.18 whilst the MSE on the test data was 0.30. The input and output set parameters after training were as shown in tables 5.8 and 5.9. The change in error with epoch number, which is shown in figure 5.14, shows that this training routine converges much more quickly to its best MSE than the example of section 5.8.3

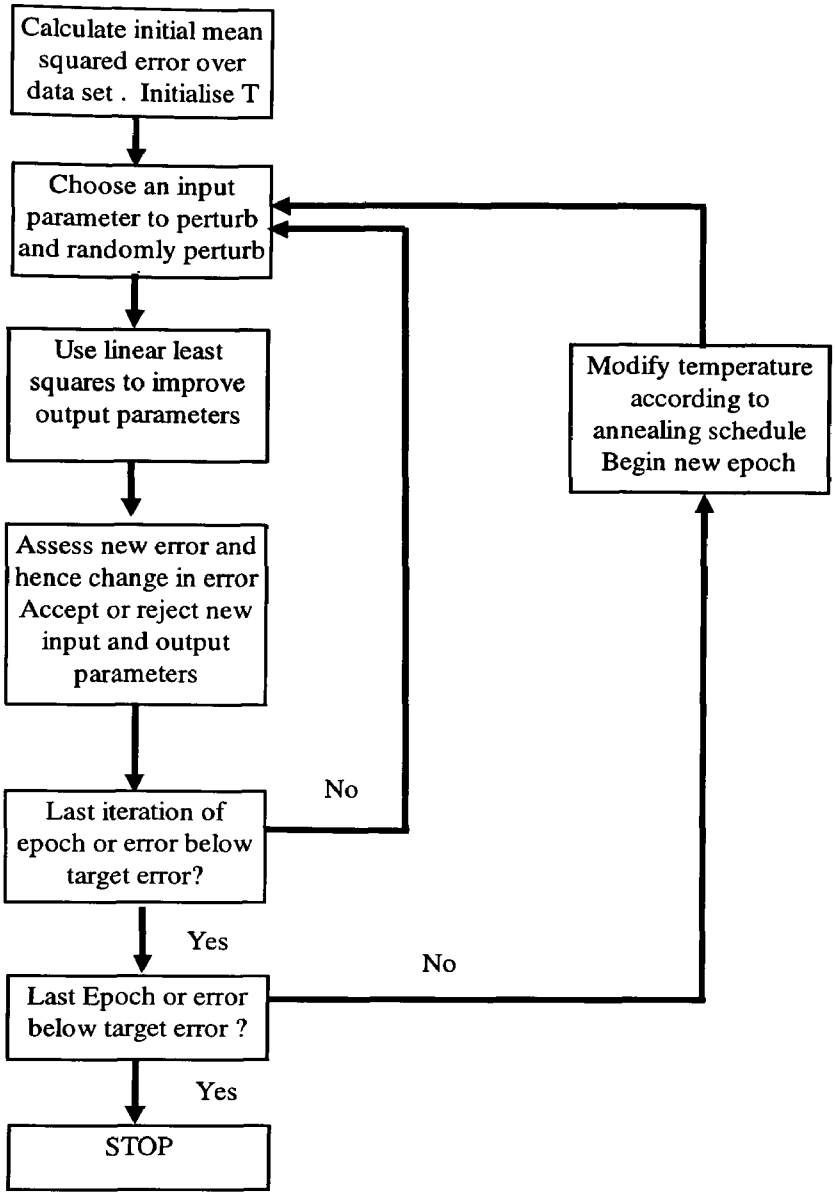

Figure 5.13: Flow diagram showing combined linear least squares and simulated annealing

	Input 1	**Input 2**
Fuzzy set 1	$\mu = 0\ \sigma = 31.4$	$\mu = 0.74\ \sigma = 51.8$
Fuzzy set 2	$\mu = 0\ \sigma = 4.89$	$\mu = 0\ \sigma = 111$
Fuzzy set 3	$\mu = 0,\ \sigma = 0$	$\mu = 0\ \sigma = 0$

Table 5.8: Parameters for Input fuzzy sets for function SQR2 after training inputs and outputs using simulated annealing and least squares

Output set r	Parameter 1 (a_1^r)	Parameter 2 (a_2^r)
1	-387	232
2	2,760	-815
3	2,510	665
4	-14,000	669
5	8,200	-2460
6	5,710	1,820
7	13,300	38,100
8	-51,100	-173,000
9	38,000	135,000

Table 5.9: Output set parameters for output fuzzy sets for function SQR2 after training inputs & outputs using simulated annealing and least squares

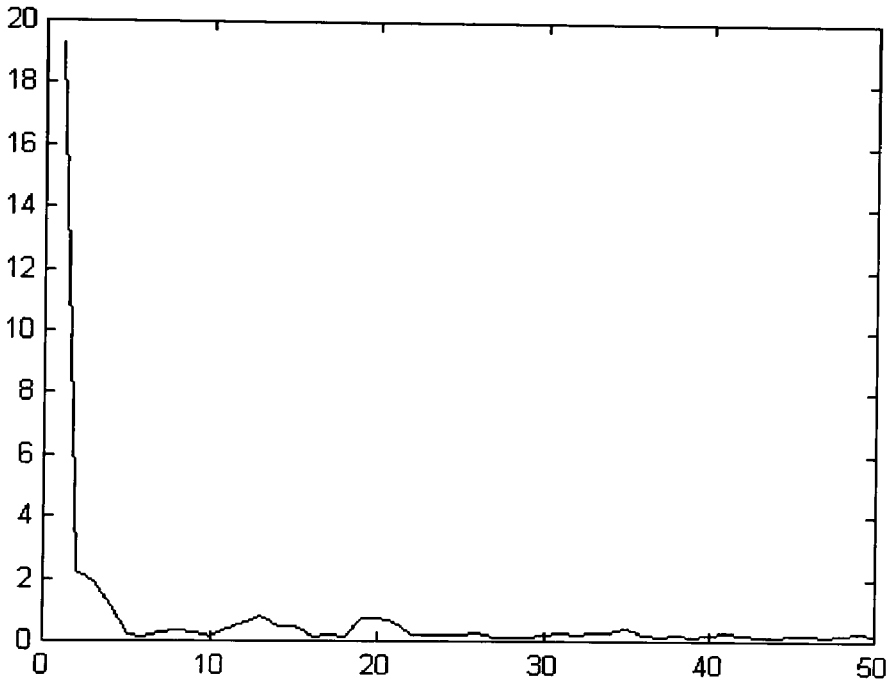

Figure 5.14: Plot of MSE versus epoch number for example of section 5.9.2

5.10 Simulated annealing combined with downhill simplex method and least squares to train input and output sets.

5.10.1 Overview of technique

This technique utilises the ability of linear least squares to optimise (at least in a least squares sense) the output parameters of the fuzzy inferencing system in a single pass. This leaves the input parameters to be optimised in some way. As discussed in section 5.7.2, simulated

annealing has the ability to search for optimal parameters without getting trapped in local minima. However it was also noted that simple simulated annealing using purely random perturbations about an existing point in error space can be inefficient. (Press *et al.*, 1994) suggest a method which combines simulated annealing with a downhill simplex technique to improve the efficiency of simulated annealing for continuous parameters. Their algorithm and code has been applied and modified in the FTEST application software for training fuzzy inferencing systems. The following subsections describe downhill simplex and its use in combination with simulated annealing. The combination of this technique with linear least squares to produce an efficient overall training algorithm for training fuzzy systems is then described.

5.10.2 Downhill simplex

The basic downhill simplex is a multidimensional function minimisation technique due to (Nelder and Mead, 1965) and described in (*op cit* Press *et al.*, 1994). The multidimensional function to be minimised is the error as a function of the fuzzy system parameters. Gradient descent techniques such as backpropagation require the evaluation of the partial derivatives of the error measure versus each parameter. The simplex method does not, however, require evaluation of the derivatives of the function that is being minimised. This is a potential advantage for its use in training a fuzzy system. In the case of fuzzy systems where the parameters to be adjusted are the input fuzzy set parameters, evaluation of these derivatives require differentiable membership functions. This is avoided by the simplex method, which only requires evaluation of the function itself rather than its derivatives.

For a fuzzy system with N inputs, F fuzzy sets per input, and Gaussian membership functions the error function is a function of $2.N.F$ variables (the parameters). In D dimensions a solid can be formed with $D+1$ vertices in such a way that if any vertex is taken as origin the vectors from

the origin to the remaining vertices span the vector space of D dimensions. Such a D-dimensional solid is called a nondegenerate simplex. In the fuzzy system error minimisation problem the simplex is formed in a $D = (2.N.F)$ space and has $D+1 = (2N.F+1)$ vertices. The co-ordinates of each vertex consist of a possible set of parameters to the fuzzy system and each vertex has an associated mean squared error over a training data set. Thus each vertex is a trial point in parameter space and has to be initialised at the start of the training.

The downhill simplex method carries out a sequence of moves that are geometrical transformations of the simplex. The moves are such that the simplex travels through the D-dimensional error space in a direction towards a minimum local to the simplex's starting point i.e. the simplex moves 'downhill'. The basic moves are called '*reflection*', '*reflection with expansion*', '*reflection with contraction*' and '*contraction*' (*op cit* Press *et al.*, 1994).

Reflection: In this move parameter co-ordinates of the vertex having the highest mean squared error are changed such that the vertex moves through the face of the simplex which is opposite it. i.e. it is reflected in that face.

Reflection with expansion: In this move if an initial reflection is successful (produces an error lower than the vertex with the best error), the reflection is extended by a factor of two i.e. the reflected vertex is moved to twice as far behind the plane of reflection as it was originally in front of it.

Reflection with contraction: This move is carried out if a reflection produces an error that is higher than that of the vertex with the second highest error. In this case a reflection with a factor of 0.5 is tried and accepted if it improves on the original error of the vertex. The factor of 0.5 implies that the reflected vertex is half the

distance behind the plane of reflection as the original vertex was in front. If after a reflection and contraction the vertex has a lower error than its original error then the move is accepted.

Contraction: This is the move carried out when all the above moves fail to reduce the error of the worst vertex. In this case all the vertices are contracted in length by a factor of 0.5 about the vertex with the lowest error.

The downhill simplex algorithm iterates over these moves in order and terminates when all the vertices have an error which is within some defined tolerance parameter of each other. The simplex contracts in directions across valleys in error space and expands along such valleys in the downhill direction. In minima the simplex keeps contracting, until all the vertices converge on the minimum point. The simplex can also manoeuvre through narrow constrictions in error space.

The downhill simplex method is an efficient method of exploring multidimensional error spaces. However it can converge on local or poor minima. It does, however, offer a useful method of generating efficient moves for a simulated annealing-based algorithm.

5.10.3 Downhill simplex combined with simulated annealing

The approach of combining the downhill simplex method with simulated annealing to produce an efficient and robust search method, which is resistant to entrapment in local minima, is due to (*op cit* Press *et al.*, 1994). In order to implement the Metropolis algorithm, rather than testing the effect on the error of a random perturbation of the system against an exponential distribution, a natural logarithm distributed random number is added to the error computed for each existing point in the simplex. Furthermore a natural logarithm-distributed random number

is subtracted from any trial points generated using the moves of the simplex described in section 5.10.2. The logarithm-distributed number is proportional to a temperature parameter, which is initialised at some high value and progressively reduced after a fixed number of iterations. The random number, *logran* , is given by:

$$logran = -T \log_e [pseudran] \qquad 5.30$$

Where *pseudran* is a uniformly distributed random number. This addition and subtraction of *logran* has the same overall effect on the evolution of the simplex as the conventional simulated annealing method has on its single search point. This effect is that some moves which result in an increase in the error of a vertex are still accepted. In (*op cit* Press *et al.*, 1994) the motion of the simplex is described as ".... a stochastic , tumbling Brownian motion ....".

The simplex method is a localised multipoint search with those points that have the lowest errors pulling the points with the worst errors in their direction. The simplex can be regarded as a set of sample points in parameter space competing to find the lowest set of energy or error states. This is similar in concept to a low population genetic algorithm. The addition of the random numbers to the error values of the vertices can be regarded as the analogue of random mutation in Genetic Algorithms.

5.10.4 Simplex, simulated annealing, and linear least squares combined algorithm

This section describes how the downhill simplex and simulated annealing method of Press *et al.* is applied in the 'FTEST' application program to train the input set parameters of fuzzy systems. In this implementation the combined simplex - simulated annealing method is also combined with the linear least squares method to produce an overall training method for Sugeno fuzzy inferencing systems. This training method can be selected in the training dialog box of FTEST under the 'Train...' options. It is called the 'AMEBSA & LLS' option there.

The name AMEBSA is used as this is the name given by Press *et al.* to their code for the simplex-simulated annealing routine described in section 5.10.3 and which is used as part of the implementation in FTEST. Whenever the error associated with a vertex is needed the AMEBSA routine calls a function 'funk' using the function call:

float funk(float* NEWparams).

In FTEST 'funk' returns the value of the mean squared error over the training data set obtained with the input parameters pointed to in the function call to 'funk', 'NEWparams', after the output parameters of the fuzzy system have been set to those giving the least squared error using the method of section 5.5.4. Thus the error associated with each vertex takes into account the optimisation possible by tuning the output set parameters. The comments of section 5.8.1 on the effects of the difference between the least squares and the least mean squares error criterion also apply to the use of the least squares method in the 'AMEBSA & LLS' approach. Figure 5.15 is the flow diagram showing how the routine AMEBSA is used in FTEST to train fuzzy systems.

5.10.5 Example: Training of input and output sets for 'SQR2' using downhill simplex, simulated annealing and least squares.

As for the previous described training techniques, approximation of the function SQR2 by a fuzzy system is taken as an example to illustrate the training technique and to allow a comparison with the other training techniques investigated. After training for 77 epochs of 100 iterations, starting with a temperature of 4 and with a cooling parameter of 0.2, an MSE of 0.009 was obtained. This is clearly an improvement on the other methods described in this chapter. The change in MSE versus epoch number is illustrated by the plot of figure 5.16. Tables 5.10 and 5.11 show the final parameter settings. The error on the verification data set was 0.032 after training.

Output set r	Parameter 1 (a_1^r)	Parameter 2 (a_2^r)
1	3.64	-411
2	-100	1030
3	287	-676
4	29.1	-21.8
5	82.7	-35.8
6	-472	95.1
7	17.1	-8.89
8	35.1	25.5
9	162	26.7

Table 5.10: Output set parameters for output fuzzy sets for function SQR2 after training inputs & outputs using the hybrid downhill simplex, simulated annealing, and least squares approach.

	Input 1	Input 2
Fuzzy set 1	$\mu = 9.27$ $\sigma = 9.35$	$\mu = 0.098$ $\sigma = 15.11$
Fuzzy set 2	$\mu = 0.98$ $\sigma = 18.23$	$\mu = 4.65$ $\sigma = 24.62$
Fuzzy set 3	$\mu = 9.56$ $\sigma = 13.89$	$\mu = 0.055$ $\sigma = 6.28$

Table 5.11: Parameters for Input fuzzy sets for function SQR2 after training inputs & outputs using the hybrid downhill simplex, simulated annealing, and least squares approach

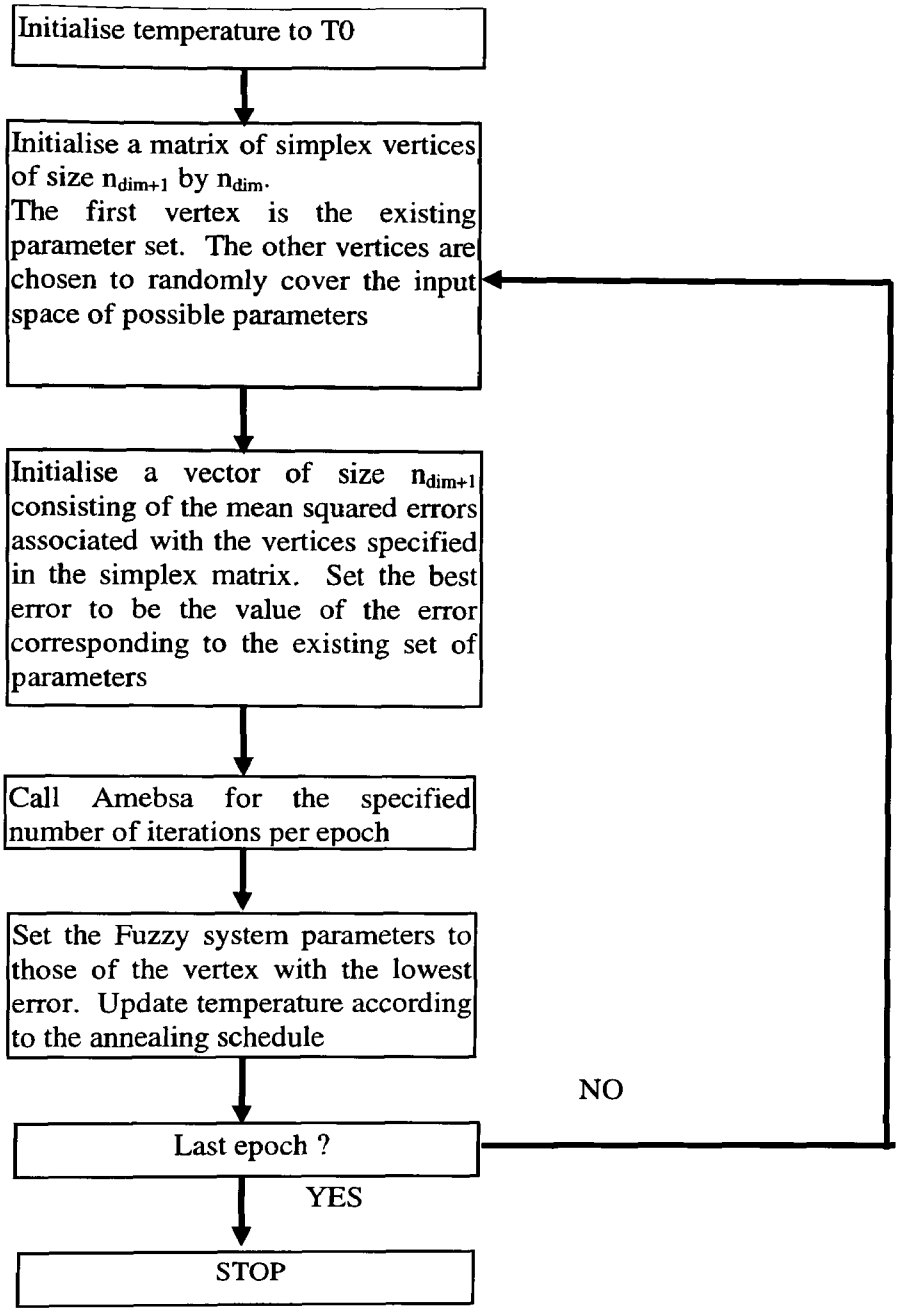

Figure 5.15: Flow diagram showing how the routine AMEBSA is used in FTEST to train fuzzy systems.

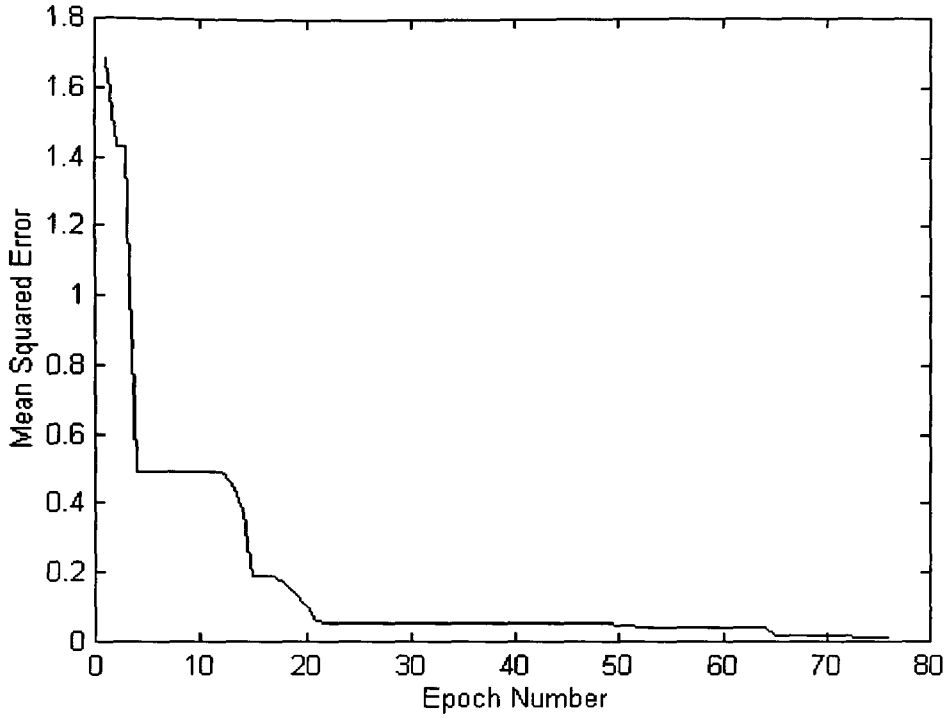

Figure 5.16: Variation of MSE versus epoch number for example of section 5.10.5

5.11 Simulated annealing and automatic rule generation for rulebase selection.

5.11.1 Non exhaustive rulebases

All the training techniques described in the previous sections have used the training to optimise input and output fuzzy set parameters. However the action of a fuzzy inferencing system also depends upon its rulebase.

In the case of the inferencing systems approximating 'SQR2', used in the training examples, the maximum possible number of rules is nine, assuming that all inputs appear in all rules. This is because there are two inputs and three fuzzy sets per input. If the further restriction is made that all the inputs are associated with the same number of fuzzy sets, there will be N_R possible rules given by:

$$N_R = F^N \tag{5.31}$$

Thus, as the number of inputs to a fuzzy inferencing system increase, the number of possible rules in a rulebase grows as the power of the number of inputs. This problem of dimensionality (Kim and Kosko, 1996) potentially causes problems in the speed of execution of a fuzzy system as the number of rules to be evaluated increases. However, a fuzzy system can be created and trained offline and the mapping from input vector space to output scalar space can be established. This mapping need not then be implemented as a program or piece of hardware evaluating rules, but can be implemented as a look up table with a trade-off between speed of execution and memory requirements. Nevertheless the power law nature of fuzzy system scaling will establish a practical upper limit to the size of workable fuzzy systems. The problem of fuzzy system scaling becomes even more acute when the scaling of the computational cost of training with number of inputs is taken into consideration. The computational cost of training is affected by the following factors and in the following ways:

The number of rules in the rulebase will affect the time taken to train a fuzzy system. This is because for every example input/output data pair all rules must be evaluated.

The number of input and output fuzzy set parameters affects the dimensionality and hence the volume of parameter space to be explored in the way described by equation 5.29. The output set parameters increase the time required to perform the singular value decomposition and are also in practice observed to cause problems of convergence in the SVD algorithm. More seriously, for the input fuzzy set parameters trained using the methods described in this chapter, the number of iterations required in each epoch should be increased in proportion to the volume of search space i.e. as the power of the number of inputs.

For equal levels of representation of possible input output combinations the number of input output pairs in the training data set should also be scaled as the power of the number of inputs.

It is not clear however that a rulebase containing all possible rules (referred-to in this thesis as the exhaustive rulebase) is inherently superior to a rulebase consisting of a subset of the exhaustive rulebase. In any case with a large number of inputs the rulebase has to be less than exhaustive in order to remain computationally tractable. A valid question for a non-exhaustive rulebase is 'which of the possible subsets of the exhaustive rulebase gives least mean squared error?'

In order to explore the effect of using a non-exhaustive rulebase, an automatic rule generating mechanism was used in conjunction with simulated annealing to select a non-exhaustive rulebase. If the maximum number of rules in a rulebase is N_R then a subset of SN_R rules was generated and the error for this particular subset was evaluated. The subset was accepted or rejected according to the usual rules for the Metropolis algorithm. Three variations of this rule-training algorithm are implemented in the 'FTEST' application software.

A further modification made to the FTEST software in order to use non-exhaustive rulebases is that a default 'ELSE' rule is included. This rule is fired if the firing weights for all the rules in the rulebase fall below a pre-set threshold. This situation arises with a trained non-exhaustive rulebase when an input is presented to the fuzzy system that was not represented in the training data. In practice with a well-configured system this default ELSE rule should be rarely fired.

5.11.2 Automatic rule generation

In order to implement rule selection using simulated annealing a function for generating individual rules is needed. Blocks of rules, forming the subsets of the exhaustive rulebase can then be generated by successive calls to the function.

Recall that all rules are of the form:

IF Ip1 is Fuzzy Set A AND Ip2 is Fuzzy set B AND

.....IpN is Fuzzy set Z THEN output is output set O

These rules can be written in the concise form:

[1,A][2,B].....[N,Z]:O

The exhaustive rulebase can be ordered e.g. for the case of three inputs and three input fuzzy sets:

[1,1][2,1][3,1]:1
[1,1][2,1][3,2]:2
[1,1][2,1][3,3]:3
...
...
[1,3][2,3][3,3]:27

Since the inputs occur in order in the rules this can be written even more concisely as:

1,1,1:1
1,1,2:2
1,1,3:3
...
...
3,3,3:27

The three digits on the left are the base three representation of the output rule number, but with the smallest digit allowed being a '1' instead of a zero and the largest digit being a '3' instead of a '2'. The method for generating the O^{th} rule then is to convert the number O-1 to its base F representation where F is the number of fuzzy sets per input and add one to each of the digits of the N digit number obtained where N is the number of inputs. This method forms the basis of the function 'Rulegen' used in FTEST for automatic rule generation. Rulegen takes the index of the rule to be generated as an argument and returns that rule in the appropriate form for

incorporation into FTEST's rulebase to the rule training routine. The code fragment for the function 'Rulegen' is shown below:

```
int* CTRNDLG::Rulegen(int rule_no,int* rulevec)
{
 int i;
 int n0=rule_no-1;
 for(i=1;i<=m_nips;i++)   // m_nips is the number of inputs
        {                 // m_nfs is the number of fuzzy sets
        int divisor=(int)floor(pow(m_nfs,m_nips-i)+0.0001);
        rulevec[i]=(int)floor(n0/divisor)+1;
        n0=n0-((rulevec[i]-1)*divisor);
        }
 return rulevec;
 }
```

The automatic rule generating mechanism is also used in FTEST to allow the user to initialise a fuzzy system with a given universe of discourse and a given number of rules, inputs, and input sets. This is the 'initialise system' option in the training dialog box. When this option is chosen FTEST generates a set of Gaussian fuzzy input sets uniformly distributed over the input universe of discourse for each input, a set of rules, and default output sets. This 'initialise system' is intended to be used to easily and quickly generate starting point fuzzy systems before applying any training algorithms.

5.11.3 Simulated annealing used to optimise non-exhaustive rulebases

Three different variants on the use of simulated annealing to optimise non-exhaustive rulebases were explored. These variants are:

> ***Variant 1***: Simulated annealing is used to select a non-exhaustive rulebase, perturbing the rules one at a time. The least squares method is also used to optimise the output set parameters before the accept or reject decision is made according to the simulated annealing algorithm. The flow diagram for this variant is shown in figure 5.17.

***Variant 2*:** In this variant whole blocks of rules within the existing non-exhaustive rulebase are changed at a time before the accept/reject decision is made. Again the output set parameters for the new rulebase configuration are also optimised using the least squares method. The size of the block of rules to be changed is chosen at random, evenly distributed from one rule to all the rules in the existing rulebase. The flow diagram for this variant is shown in figure 5.18.

***Variant 3*:** This variant is the same as *variant 2* but in addition the amebsa routine is called with a very low temperature to locally optimise the input fuzzy sets for the new rule configuration. The least squares method is used to optimise the output set parameters as before. By using a low temperature setting the amebsa routine is effectively reduced to a downhill simplex method. Also only one epoch of a fixed number of iterations is used in the call to 'amebsa'. This variant is a compromise of the possible but computationally longer approach of carrying out a full simulated annealing optimisation of input and output set parameters for every possible rule configuration. The flow diagram for this variant is shown in figure 5.19.

A different approach to rulebase training is suggested in a recent report on training a Mamdani fuzzy system (Garibaldi and Ifeachor, 1999). In this approach the choice of rules in the rulebase is treated as one of a set of discrete variables which also include other structural parameters of a fuzzy system. A random choice is then periodically made in the training algorithm as to whether to train discrete or continuous parameters for a fixed number of iterations.

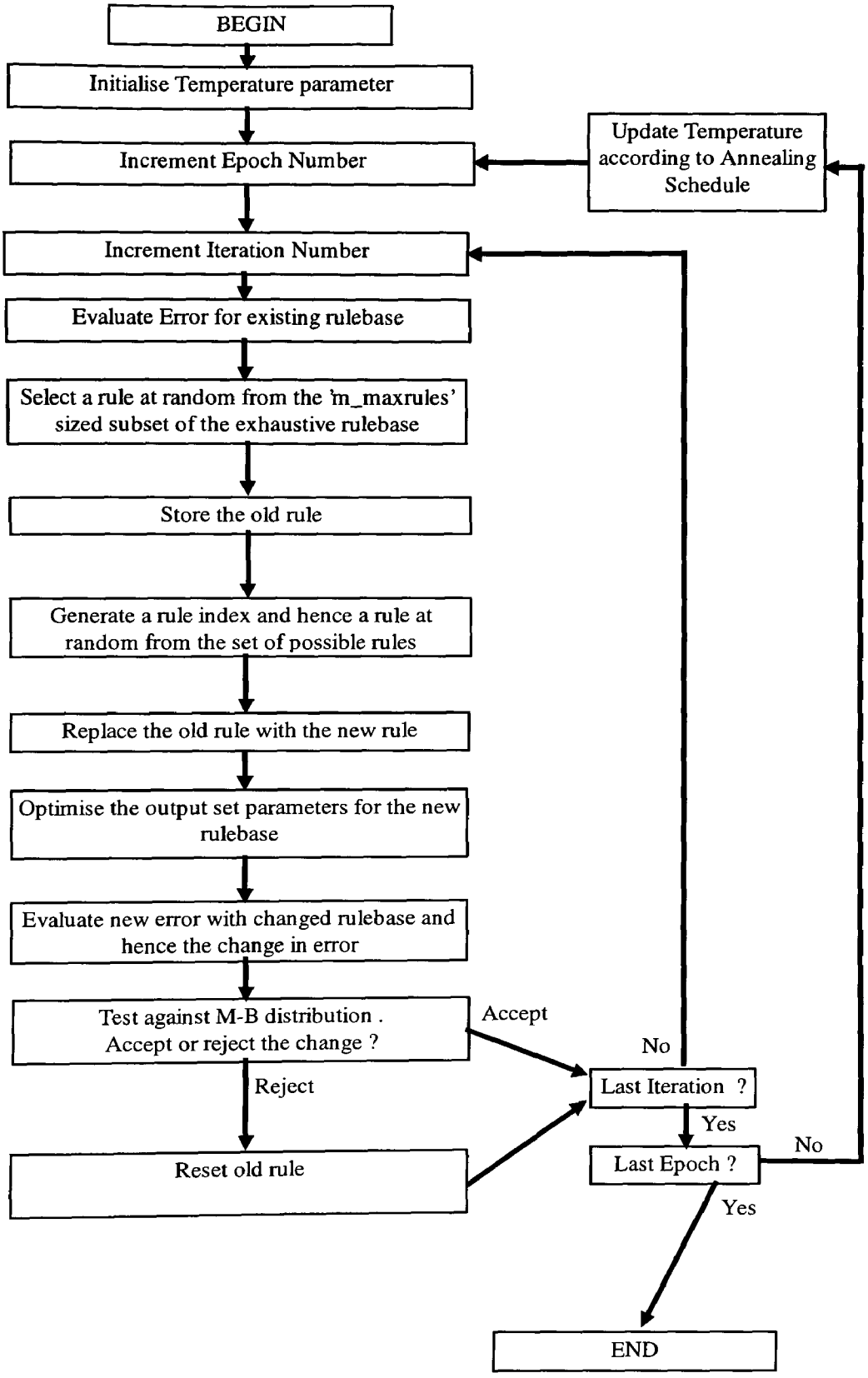

Figure 5.17: Flow diagram for rulebase training using Variant 1

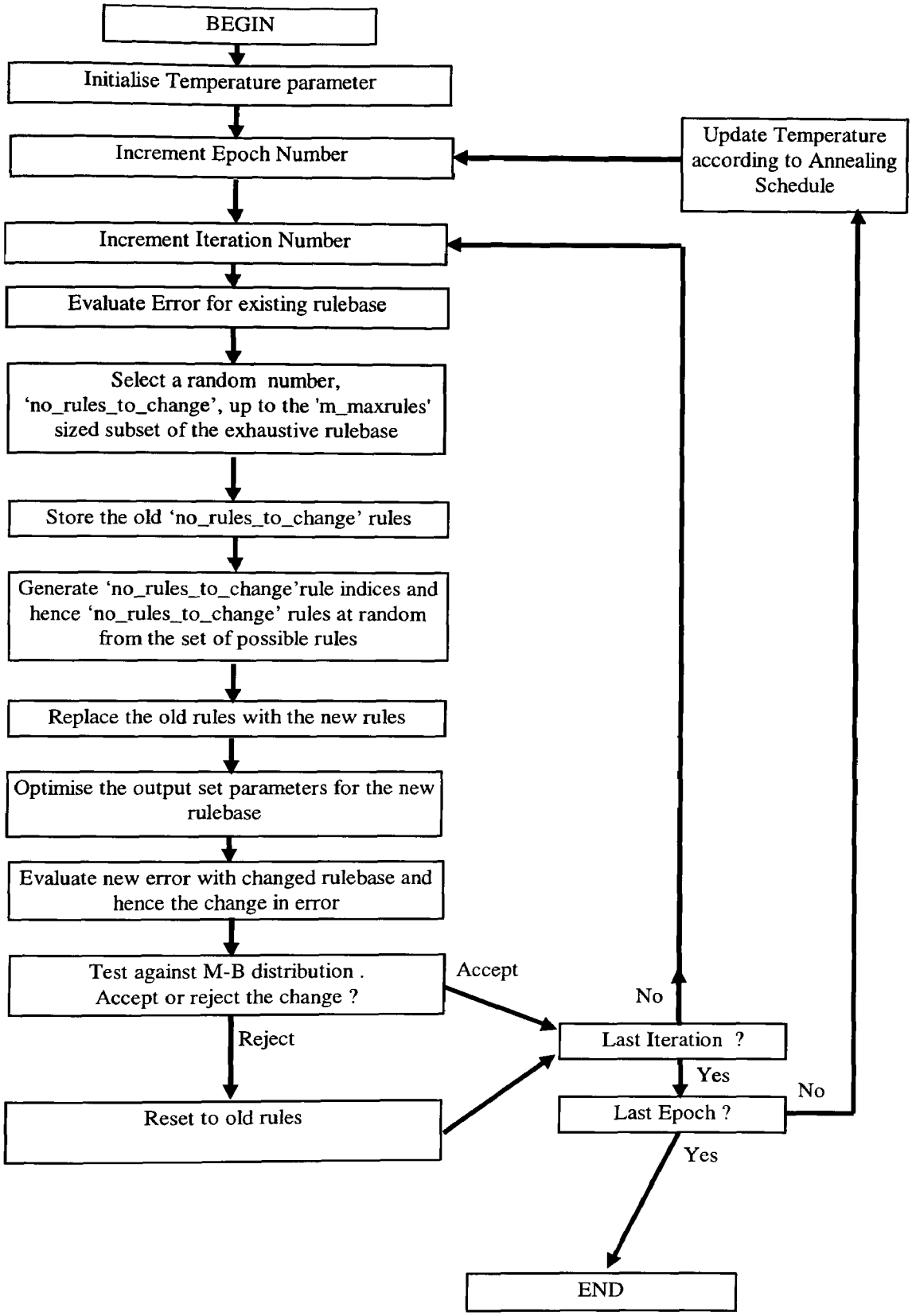

Figure 5.18: Flow diagram for rulebase training using Variant 2

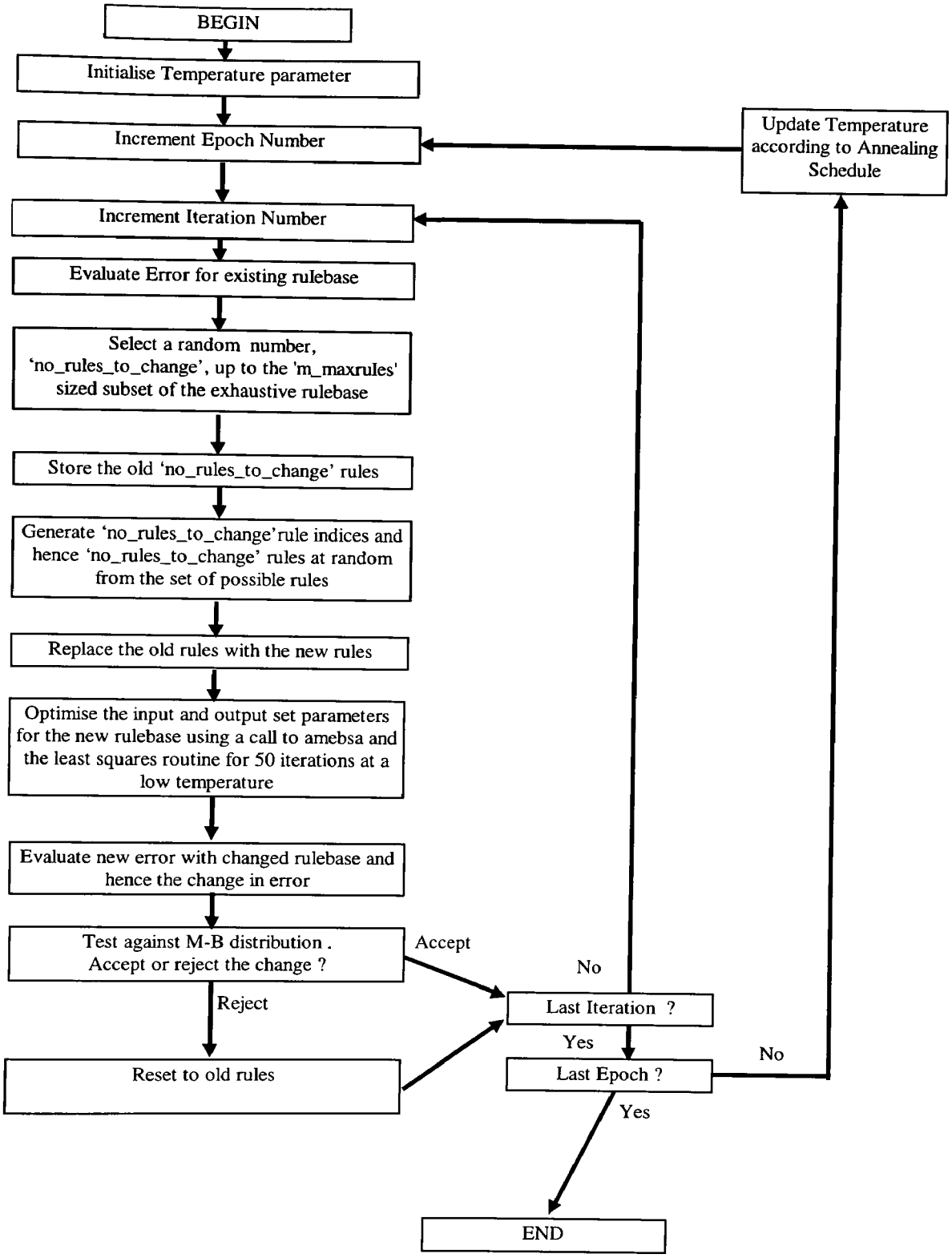

Figure 5.19 Flow diagram for rulebase training using Variant 3

5.11.4 Examples of rulebase training

The function SQR2 was again used to test the rulebase training routines. In this case, however, the 'initialise system' option of FTEST was used to set up a fuzzy system with three input fuzzy sets per input but with only six out of the possible nine rules. The task to which the three rulebase training variants were each applied was to select an optimal set of six rules with their associated output sets. In the case of variant 3 some optimisation of the input sets is also carried out before selecting or rejecting a particular rule combination.

After initialising the fuzzy system in FTEST the MSE over the training data set was 5,644, which reduced to 2,648 after applying the least squares routine to optimise the output set configuration. This raw system was saved and used as the start point for all the rulebase training routines tested. The hybrid simplex, simulated annealing, and linear least squares algorithm described in section 5.10.4 was also applied to the raw system so that a comparison could be made between training a non-exhaustive rulebase system with no rule selection and the rule-selecting methods of variants 1 to 3. The hybrid simplex, simulated annealing, and linear least squares algorithm achieved a MSE of 0.6 on the training data and 4.7 on the test data after training for 100 epochs of 100 iterations.

When applied to the raw fuzzy system, both variant 1 and variant 2 of the rulebase training algorithm consistently achieved a minimum MSE of 192 on the training data and 299 on the test data. Moreover the set of rules chosen by both variants was also the same. This suggests that for this problem, with the particular choice of input sets made by the FTEST initialising routine, this was the optimum subset of six rules, and that neither training method variant offered an advantage over the other. Typical plots of MSE versus epoch number for variants 1 and 2 for this example are shown in figures 5.20 and 5.21.

The MSE achieved by variants 1 and 2 on this problem is poor by comparison with that achieved by the hybrid simplex, simulated annealing, and linear least squares algorithm, since the input sets are not optimised by variants 1 and 2. When, however, the hybrid simplex, simulated annealing, and linear least squares algorithm was applied to the system generated by variants 1 and 2 a MSE of 0.046 was achieved on the training data and 0.123 on the test data.

Variant 3 of the rulebase training routines improved the MSE of the raw six-rule version of SQR2 to a MSE of 2.8 on the training data and a MSE of 3.25 for the test data. The plot of MSE versus epoch number for this example is shown in figure 5.22. Since variant 3 locally optimises the input sets before accepting or rejecting a particular rule configuration, the variant 3 solution for the best rule configuration might be expected to be different from those of variants 1 and 2. However, for this example, the rule configuration chosen by variant 3 differed in only one rule from that chosen by variants 1 and 2.

When the six-rule fuzzy system for SQR2 generated by variant 3 was trained using the hybrid simplex, simulated annealing, and linear least squares algorithm, the MSE was further reduced to 0.053 on the training data and 0.07 on the test data.

5.11.5 Discussion of rulebase selection strategies

The results of the examples discussed in section 5.11.4 suggest that for a non-exhaustive rulebase, which combination of rules is chosen influences the ultimate MSE that can be achieved, even with subsequent optimisation of the input sets. Moreover the results suggest that if, after rulebase training, the input sets are subsequently trained using an approach like the hybrid simplex, simulated annealing, and linear least squares algorithm, there is little difference between the three possible approaches to rulebase training discussed in section 5.11. However, there is scope for further work to establish whether this result extends to fuzzy rulebase training in general.

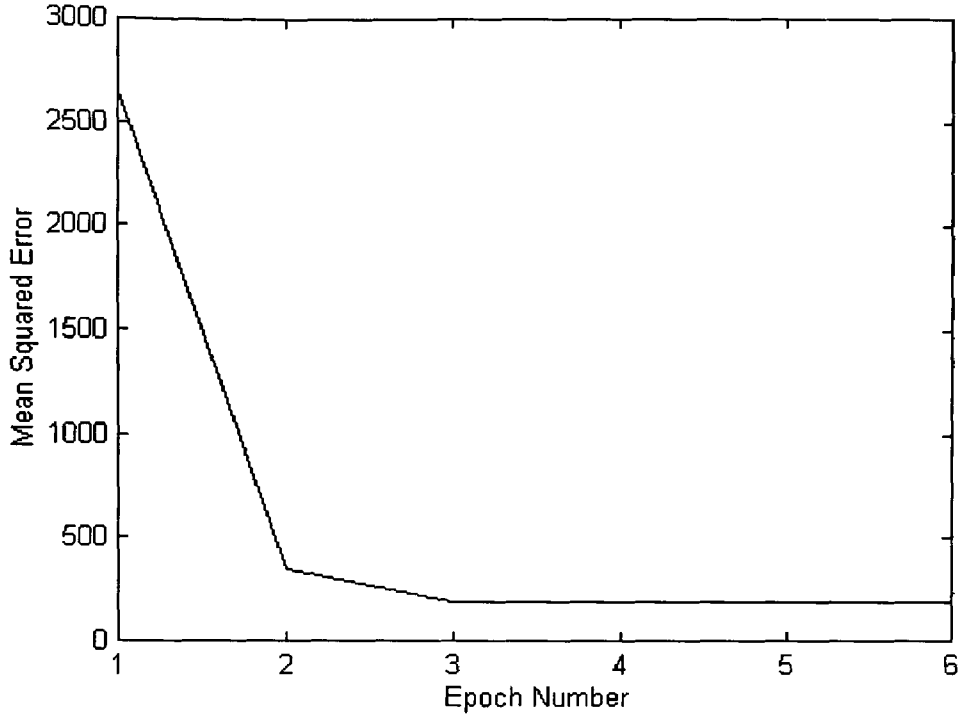

Figure 5.20: Plot of MSE versus epoch number for variant 1 for example of section 5.11.4

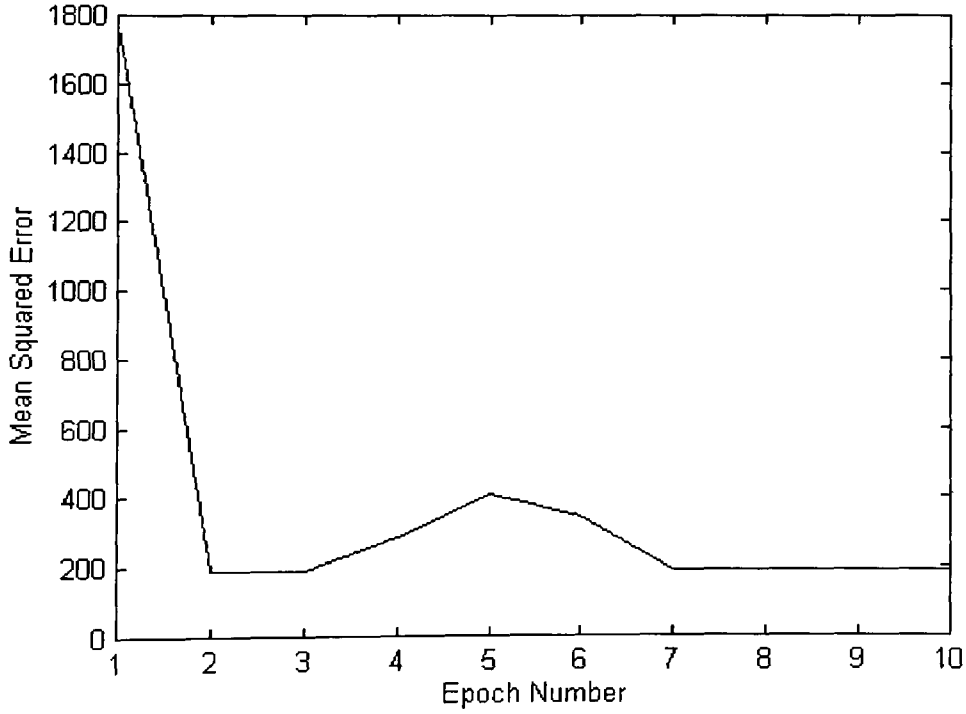

Figure 5.21: Plot of MSE versus epoch number for variant 2 for example of section 5.11.4

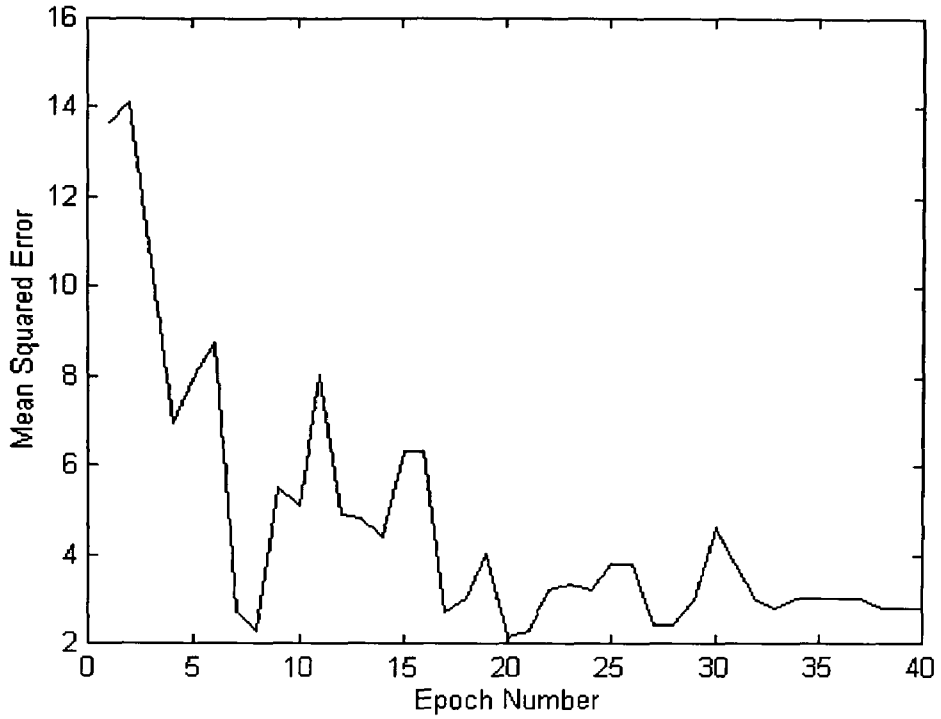

Figure 5.22: Plot of MSE versus epoch number for variant 3 for example of section 5.11.4

5.12 Summary of Chapter 5

Chapter 5 has presented a brief overview of fuzzy sets, fuzzy logic, and Mamdani and Sugeno fuzzy inferencing systems. The chapter has introduced the FIR type of first order Sugeno system that is investigated in Chapter 7 as a basis for filtering signals, such as depth maps, corrupted by mixed Gaussian and impulsive noise. After a brief description of the FTEST software that is used in the work described in this thesis to investigate and generate fuzzy systems, the chapter has introduced six related approaches to fuzzy system training. These are based on combinations of a linear least squares method, the simulated annealing algorithm and the simplex method. The remainder of the chapter has been concerned with testing these six approaches using a simple function approximator example. When used to train fuzzy systems with a fixed rulebase, the hybrid simplex, simulated annealing, and linear least squares algorithm produced the best results. The need for non-exhaustive rulebases because of the

problem of rulebase explosion was introduced and three approaches to fuzzy system training with rulebase selection were suggested and tested. The training approaches to Sugeno system training introduced in this chapter and which form a part of the contribution made by the thesis, are used in the training of the filters discussed in Chapter 7.

5.13 References

(Chiang *et al.*, 1992). Chiang W-C, Gutierrez G.J., and Kouvelis P. "Simulated Annealing and Tabu Search" in Intelligent Design and Manufacturing, Edited by Kusiak A. pp 673-700 Wiley 1992.

(Bandemer and Gottwald, 1995) Bandemer H and Gottwald S. "Fuzzy Sets, Fuzzy Logic Fuzzy Methods with Applications." Wiley, 1995.

(Epp 1990) Epp S.S. "Discrete Mathematics with applications." PWS Publishing Company. 1990.

(Garibaldi and Ifeachor, 1999) Garibaldi J. M. and Ifeachor E.C. "Application of Simulated Annealing Fuzzy Model Tuning to Umbilical Cord Acid-Base Interpretation." IEEE Transactions on Fuzzy Systems Vol 7. No 1 pp 72- 84. Feb 1999.

(Goldberg, 1989) Goldberg D.E. "Genetic Algorithms in Search, Optimisation, and Machine Learning." Addison-Wesley, 1989.

(Holmblad and Østergaard, 1982) Holmblad L.P. and Østergaard J.J "Control of a cement Kiln by fuzzy logic" Fuzzy Information and Decision Processes. North-Holland publishing Company pp 389-399, 1982.

(Jang, 1993) J-S R Jang "ANFIS: Adaptive-Network-Based Fuzzy Inference System." IEEE Transactions on Systems Man and Cybernetics Vol 23 No 3 pp 665-684 May/June 1993

(Kirkpatrick *et al.*, 1983) Kirkpatrick S., Gelatt C. D., and Vecchi M. "Optimisation by Simulated Annealing." Science. Vol 220, Issue 4598. pp 671-680. May 1983.

(Kirkpatrick, 1984) Kirkpatrick S., "Optimization by Simulated Annealing: Quantitative Studies." Journal of Statistical Physics, Vol34 Nos 5/6 pp 975 – 986. 1984.

(Larsen, 1980) Larsen R. "Industrial applications of fuzzy logic control" International Journal of Man-Machine Studies Vol 12 pp 3-10 1980.

(Lawson and Hanson, 1974) Lawson C.L and Hanson R.J. "Solving Least Squares Problems." Prentice Hall, 1974.

(Lee, 1990(b)). Lee C.C. "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part II" IEEE Transactions on Systems Man and Cybernetics. Vol 20 No 2 pp 419-435. March/April 1990.

(Mamdani and Assillian, 1975) Mamdani, E. H. and Assillian, S. "An experiment in linguistic synthesis with a fuzzy logic controller." International Journal of Man-Machine Studies. No 7 (part 1) pp 1-13. 1975.

(Mamdani, 1974) Mamdani, E. H. "Application of fuzzy algorithms for control of simple dynamic plant." Proceedings of the IEE- Control and Science; Vol 121 No12. pp 1585-1588. December. 1974.

(Mandl, 1973) Mandl F "Statistical Physics" Manchester Series on Physics Wiley 1973

(Mendel, 1995) Mendel J.M. "Fuzzy Logic Systems for Engineering:A Tutorial." Proceedings of the IEEE Vol 83 No 3 pp 345-377 March 1995.

(Metropolis *et al.*1953) Metropolis N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H., Teller E. "Equation of State Calculations by Fast Computing Machines" Journal of Chemical Physics Vol 21, No 6. pp 1087-1092. June 1953.

(Nelder and Mead, 1965) Nelder J.A, and Mead R. "A simplex method for function minimization" Computing Journal. Vol 7 pp 308-313 1965.

(Press *et al.*, 1994) Press W.H., Teukolsky S.A., Vetterling W.T., and Flannery B.P. "Numerical Recipes in C The Art of Scientific Computing"2nd Edition C.U.P. 1994.

(Takagi and Sugeno, 1983) Takagi T and Sugeno M, "Derivation of Fuzzy Control Rules from Human Operator's Control Actions." Proceedings IFAC Symposium on Fuzzy Information Knowledge Representation and Decision Analysis , pp 55-60. 1983.

(Vanderbuilt and Louie, 1984) Vanderbuilt D. and Louie S. G. "A Monte Carlo Simulated Annealing approach to Optimisation over Continuous Variables." Journal of Computational Physics Vol. 56 pp 259-271, 1984.

(Zadeh, 1965), Zadeh L.A. "Fuzzy Sets" Information and Control Vol 8, pp 338-353 1965

(Zadeh, 1973) Zadeh L.A. "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. " IEEE Transactions on Systems Man and Cybernetics. Vol 3 No1 pp 28-44 January 1973

Chapter 6: Fuzzy logic-based filters

6.1 Introduction

This chapter describes the development of fuzzy logic-based filters for filtering signals corrupted by mixed Gaussian and impulsive noise, such as the depth maps described in Chapter 3. A review of previous work on the use of fuzzy logic in filtering problems is also presented in this chapter and a taxonomy of such fuzzy filters is proposed. A fuzzy logic-based filter using a zero order Sugeno network (Takagi and Sugeno, 1983) is described and applied to the task of smoothing disparity maps derived from real image sequences. A universal fuzzy filter architecture based on the FIR type (see section 5.3.2) first order Sugeno fuzzy inferencing system is then presented. The performance of this filter architecture can be optimised using the techniques described in Chapter 5. The contributions of this chapter are the proposed fuzzy filter taxonomy, the application of the zero order filter to disparity map smoothing, and the proposed use of the FIR type Sugeno network as a fuzzy filter architecture.

6.2 Motivation for the use of Fuzzy Logic in filtering

The use of fuzzy logic in tackling the problem of filtering dense depth maps produced by correlation based matchers is philosophically motivated by the ill-posed nature of the problem of establishing correspondence using correlation which was discussed in section 3.6. As discussed there, the classical way of dealing with ill-posed problems is to impose additional constraints to the problem formulation such that the best solution is chosen given uncertain and ambiguous measurements. These additional constraints express *a priori* assumptions about the form of

acceptable solutions. For certain constraints expressed in a variational form, a known linear or nonlinear filter can be identified. Thus for these cases the mapping from input to output carried out by a filter is identical to applying the prior constraint corresponding to that filter. Because of their universal function approximation property (Kosko, 1992), (Wang, 1992), (Wang and Mendel, 1992) fuzzy inferencing systems can be used to approximate any mapping from an input vector to a single output. The use of fuzzy inferencing systems as general input-output mappers or general filters allows the regularising *a priori* assumptions to be potentially explicitly embodied in the fuzzy rule base and the input and output membership functions. Neural networks can also be used as general input-output mappers (Hornik *et al.*., 1989) and neural networks have been used as the basis for image processing filters (Pugmire *et al.* 1995). The mapping performed by a neural network is entirely derived from training using input-output data pair. Thus, the *a priori* assumptions being applied are only implicit in this data and not explicit. However, knowledge in the form of input-output pairs can also be useful in deriving filter mappings. Fuzzy inferencing systems possess the twin advantages of being trainable like neural networks as well as being derivable using prior knowledge. This makes them attractive in principle as a general filtering method, and gives them an advantage over neural network based filtering. Filters based on fuzzy inferencing can also potentially make explicit use of known filtering techniques.

6.3 A taxonomy of Fuzzy filters

In the discussion in this chapter fuzzy filters are divided into two classes, direct and indirect acting. The first class of filters which are examined here are termed 'indirect acting' because the fuzzy system is used to determine the parameters of a conventional filter system. If this conventional filter system is viewed as a Black box which has external control knobs, the indirect fuzzy systems take the place of an operator manipulating the control knobs. The control knobs are manipulated

by a fuzzy expert system, which is external and separated from the filter system. The action of the fuzzy system on the raw input signal is 'indirect'. This is the most commonly used approach to applying fuzzy inferencing systems to filtering problems.

The distinction between direct and indirect filters is a fundamental one. A direct filter architecture based on Mamdani Inferencing (Mamdani, 1974) is a pure fuzzy system. In such a pure system, input numerical data is turned into linguistic (fuzzy) data by a fuzzifier stage. The fuzzy system infers a linguistic output from the fuzzy input data according to its internal rulebase and inferencing method. Finally, the linguistic output data is converted into crisp numerical data by the defuzzifier stage. This type of filter architecture is illustrated in figure 6.1

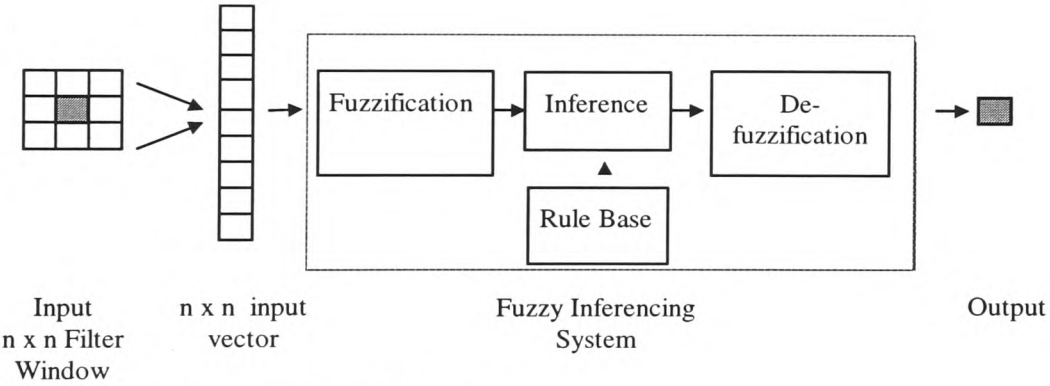

Figure 6.1: Direct acting fuzzy filter architecture.

Because of its purity as a fuzzy system the design and analysis of a direct fuzzy filter can draw on the body of knowledge concerning fuzzy calculus. However the direct acting architecture with Mamdani inferencing requires care in the fuzzification stage to ensure unique signal representation as a fuzzy set and to avoid 'fuzzy aliasing' (deOliveira, 1996). The necessary and sufficient condition for this is discussed in (*op cit* deOliveira, 1996) and is that the mapping performed by the

fuzzification stage from crisp numerical input to fuzzy set is injective. If this condition is met then two different input vectors always result in a different fuzzy representation. If two different input vectors result in the same fuzzy representation then, according to (*op cit* deOliveira, 1996), fuzzy aliasing takes place. Although (*op cit* deOliveira, 1996) does not state this, there is an underlying assumption that the fuzzy system is memoryless, which is normally the case. Ensuring that this condition is met complicates the training of a fuzzy system using numerical input-output data unless the sufficient condition in (*op cit* deOliveira, 1996) of linearity through the direct fuzzification/defuzzification process is met. By contrast, an indirect fuzzy system always acts on the data by controlling a conventional filter stage or selecting between such stages. Thus, non-uniqueness of the signal representation in the fuzzy inferencing system does not result in the same output for different inputs. Different inputs can result in the same type of filter *action* however. Similar filter action for different inputs is often perfectly acceptable. This distinction between direct and indirect fuzzy systems is not drawn in (*op cit* deOliveira, 1996). Filters based on the Sugeno inferencing system, are hybrid systems in the direct/indirect taxonomy. They do not suffer from the fuzzy aliasing problem of direct Mamdani based filters, but as is shown in section 6.6, they encapsulate the whole of the filter within a single fuzzy inferencing system. This encapsulation of the filter action facilitates the training of such filters with input-output training data.

6.4 Review of work on fuzzy-based filters

The majority of the papers on fuzzy-based filters discussed in this section describe filters that are based on an adjusted weighted mean filter. In this type of filter, having a filter window of size 2N+1 the output value $y[n]$ for a particular sampling point $x[n]$ indexed by 'n' is given by:

$$y[n] = \frac{\sum_{k=n+N}^{k=n-N} h[n-k].x[n-k]}{\sum_{k=n+N}^{k=n-N} h[n-k]} \qquad 6.1$$

This is very similar to a conventional linear transversal or finite impulse response filter except that the filter weights are determined for each value of 'k' as some function of the input data in the window. This function is achieved using some form of fuzzy inferencing system in many of the indirect fuzzy filters described in the papers discussed in this section

(Arakawa and Arakawa, 1991) is an early paper which describes the use of fuzzy logic-based filters to cope with signals which have non-stationary statistics. The indirect filter is an adjusted weighted mean type in which the weights are determined by the fuzzy stage. In this case the fuzzy stage simply consists of a stepwise approximation to a nonlinear fuzzy membership function. The nonlinear membership function defines the set whose elements are described by: "Is a member of a region having the same statistics as the centre value in the filter window". The membership function is a function of the absolute difference from the centre value in the filter window. The stepwise approximation is obtained by training using a representative training data set and a gradient descent method. The filter is applied to a one-dimensional signal corrupted with Gaussian noise and its performance compared with a moving average filter. A performance comparison is also made with an adjusted weighted average filter in which a crisp rather than fuzzy decision is made as to whether an input value belongs to the same type of signal region as the centre value in the filter window. The fuzzy filter outperformed both these filters and was reported to be more robust than the crisp adjusted weighted mean filter.

(Kwan and Cai, 1993) describe two types of filter based on a 'fuzzy concept'. Both types of filter use the adjusted weighted mean method of filtering. The weights applied to each sample in the filter window are a nonlinear function of the input data, the window median, and the maximum and minimum value in the filter window. As in (*op cit* Arakawa and Arakawa, 1991) there is no explicit rulebase and fuzzy inferencing to derive the filter weights. The method of deriving the filter weights is akin to a Fuzzy Associative Memory (Kosko, 1992). However, the basic filter structure is used in many other designs of fuzzy filters.

(Taguchi *et al.*, 1994) describe a fuzzy inferencing system which is used to determine the weighting coefficients of a 5x5 two-dimensional weighted mean filter for grey scale image restoration. This filter is therefore of the indirect type. The Mamdani fuzzy inferencing system used in (*op cit* Taguchi *et al.*, 1994) uses the distance from the centre of the window and the difference in grey scale value from the centre pixel's grey scale value as input variables and produces the weight for each pixel in the window as output. The fuzzy system is optimised by hand using the improvement in signal to noise produced by the filter as a metric. A remark is made at the end of the paper that the difference from the filter window median rather than the centre pixel value as input variable would make the filter more effective in removing impulsive noise.

(Takashima *et al.*, 1995(a)) and (Takashima *et al.*, 1995(b)) are a continuation of the work of (*op cit* Taguchi *et al.*, 1994). (*op cit* Takashima *et al.*, 1995(a)) adds 'macroinformation' based on the local image statistics as an additional input variable to the fuzzy system. This paper also adopts zero order Sugeno type inferencing. The training of the fuzzy set parameters is based on a backpropagation type of rule. The purpose of the additional observation variable is to ensure that in areas of the image which do not contain discontinuities, the output weighted mean filter being

controlled by the fuzzy filter adopts equal weights for all pixels. Thus in these image areas the filter approximates an optimal filter for Gaussian noise. It is assumed that no impulsive noise is present. The 'macroinformation' variable *k(i,j)* measures the presence or absence of discontinuities within a filter window by comparing the sample variance within the filter window with the (assumed known) Gaussian noise variance. Specifically:

$$K(i,j) = \frac{Var(i,j) - \sigma_n^2}{Var(i,j)} \quad Var(i,j) > \sigma_n^2$$
$$K(i,j) = 0 \quad Var(i,j) <= \sigma_n^2 \qquad 6.2$$

The use of this 'macroinformation' derived from local statistics is based on the work of (Lee, 1980). (*op cit* Takashima *et al.*, 1995(a)) demonstrate an improvement over the two input filter of (*op cit* Taguchi *et al.*, 1994), the mean, the median, and the modified trimmed mean (Lee and Kassam, 1985) filters tested on one and two-dimensional signals corrupted by Gaussian noise. (*op cit* Takashima *et al.*, 1995(b)) also describe an indirect fuzzy filter which determines the weights of a weighted mean filter. In this case, the weight-determining fuzzy system is in two stages. The second stage determines the weight assigned to a pixel as in (*op cit* Taguchi *et al.*, 1994) and (*op cit* Takashima *et al.*, 1995(a)), based on the difference from a reference value and the distance from the filter window centre. The reference value is derived from the first fuzzy stage. The purpose of this first stage is to determine a reference which is undisturbed by impulsive noise. The first stage incorporates a median filter as a pre-filter to achieve this. The results of applying the two stage fuzzy filter to images corrupted by both Gaussian and mixed Gaussian and impulsive noise show an improvement over the performance of mean, median, and modified trimmed mean filters. A direct

comparison with the filters of (*op cit* Taguchi *et al.*, 1994) and (*op cit* Takashima *et al.*, 1995(a)) is not presented.

(Peng and Lucke, 1994) describe a filter for mixed noise removal in images which uses the concept of a fuzzy set to determine the weights applied to each pixel in a weighted mean filter. The weight is determined directly from the degree of membership of each pixel in a set 'difference from centre pixel value'. The membership function parameters of this set are determined from training data using an iterative gradient estimate technique. In order to eliminate impulsive noise any pixel which is 'too different' from the filter window median is removed and replaced by the median. 'too different' is presumably defined by a hard thresholding operation. This paper does not explicitly use fuzzy inferencing. (Peng and Lucke, 1995) develop this filter design into a 'Multi-Level Adaptive Filter'. This filter is derived from the observation that for the filter of (*op cit* Peng and Lucke, 1994) the optimal shape of the weight-determining fuzzy set's membership function is the same for different image and noise statistics but the scaling (width) of the optimal set is different for different image and noise statistics. An additional fuzzy inferencing stage is therefore added to determine a scaling factor for the weighting fuzzy set membership function. This Mamdani inferencing stage uses the local variance of the filter window as an observation variable.

A novel approach to using fuzzy methods of modifying classical filters is taken in (Hsiao and Lai 1995). Here a least squares filter is modified by a fuzzy inferencing system. A conventional least squares algorithm is first run on the data. The residuals (differences between the observed data and the estimated data multiplied by the measurement matrix) of the least squares algorithm are then taken as observation variables into the fuzzy inferencing system. The triangular fuzzy sets of the fuzzy system have their universe of discourse and scaling set depending on the mean and standard deviation of the residuals. The fuzzy system calculates a weighting to be applied to the residuals.

Those residuals which are higher relative to the mean of all residuals are given a lower weighting. A new observation is then derived from the original observation by subtracting the original residual and adding the new weighted residual. The least squares algorithm is then re-run to obtain a modified least squares estimate. The overall effect is that the fuzzy system is used to classify the reliability of the input data and thus re-weight the data input to the least squares filter based on the size of the residuals. A similar approach is also described for modifying a recursive least squares filter and a Kalman filter.

In (Yang and Toh 1995) a fuzzy inferencing method is applied to a 'Multilevel Median Filter' (MLMF) to derive an 'Adaptive Fuzzy Multilevel Median Filter' (AFMMF). In the MLMF four one-dimensional median filters oriented vertically, horizontally, and in the two diagonal orientations within a filter window centred on pixel (i,j) is taken. The maximum and minimum of these medians is taken and the median of these two values and the value of the centre pixel (i,j) is taken as the final output of the MLMF filter. The MLMF has a better ability than the median filter to preserve line structures. The AFMMF is better able than the MLMF to remove noise that appears as 'short line like noise'. A fuzzy associative memory (*op cit* Kosko, 1992) is used to derive a confidence value for each output of the oriented one-dimensional median filters. The outputs of the two one-dimensional filters which have the best confidence values are added to the final overall median filter.

(Russo, 1996) (an earlier very similar paper is (Russo, 1993)) uses a fuzzy system to determine a correction to the centre value in a filter window. The input to the correction determining stage is the difference from centre value for each sample in the filter window. A modification is also made to a type of inferencing mechanism which is called 'Fuzzy Inference Ruled by Else-Action' or

'FIRE' (Russo and Ramponi, 1994). The essence of this inferencing technique lies in partitioning the fuzzy rulebase into sub-rulebases. Within each sub-rulebase, the consequent output set is the same for all rules and is mutually opposed to the consequent sets of the other sub-rulebases. There is also a single 'ELSE' rule, which activates a default action if none of the sub-rulebases are activated. Within each sub-rulebase the MIN operation is used for the AND operator and the MAX operator is used to aggregate the rules. Correlation product inferencing is then used to aggregate the output of all the sub-rulebases and the ELSE rule. The use of the 'MAX' operator for aggregation within the sub rulebase and correlation product to aggregate all the sub rule-bases and ELSE rule gives the fuzzy system a multi-layer architecture. (Russo, 1996) demonstrates the removal of mixed impulsive and uniformly distributed noise from a simulated one-dimensional signal and a real two-dimensional image. The problem of rulebase explosion is also mentioned as the support neighbourhood of the filters is increased and cascading fuzzy filters is suggested as a solution.

An indirect approach, described in (Arakawa, 1996), is to use a fuzzy inferencing system to perform a fuzzy or soft selection between the output of a median filter and raw unfiltered data. This is called a 'conditional median' filter in the paper. The fuzzy system is used to identify the presence or absence of impulsive noise in a flexible way. The first observation variable used is the absolute difference between the centre pixel and the median. If this observation variable is high then impulsive noise is judged to be present. A second observation variable is the mean of the differences between the centre pixel and the two pixel values closest to the centre pixel. Again, a high value of this observation variable points to the presence of impulsive noise. A two-dimensional function of the two observation variables yields a coefficient 'μ' which is used to determine the filter output as:

$$output(i, j) = median(i, j) + \mu.[input(i, j) - median(i, j)] \qquad 6.3$$

The two-dimensional function is built up of piecewise steps, which are trained using a gradient descent algorithm. This constrains the shape of the membership function surface less than if a smooth nonlinear function were used. The results of applying this type of filter to images containing fine line-like structures corrupted by impulsive noise shows an improvement over both conventional and conditional median filtering. In this paper, the nonlinear mapping from the two input observation variables to the weighting coefficient μ is derived directly from training, rather than by using the formalism of conventional fuzzy inferencing systems.

In (Mancuso *et al.*, 1996) a very similar approach to that of (*op cit* Arakawa, 1996) is taken to reduce impulsive noise in television signals. In (*op cit* Mancuso *et al.*, 1996) this is referred to as a 'decision directed filter'. Within the taxonomy of introduced in this chapter this filter is also an indirect fuzzy filter. In order to achieve the real time processing needed for television signals the fuzzy system is broken down into three layers of small fuzzy systems. The first layer produces a fuzzy possibility measure of the presence of impulsive noise in each of four oriented one-dimensional windows within the overall two-dimensional window. The possible existence of impulsive noise is determined by the differences in value between the centre pixel and its two neighbours. The second and third layers disambiguate this possibility from the existence of a small line-like structure in the image. The fuzzy inferencing sub systems use 'Min' for the AND operator and use centroid defuzzification. Each sub-system has only one rule and an ELSE clause. The structure of the overall system has therefore close similarities to that of (Russo, 1996).

In contrast to all the above fuzzy filters, a direct acting fuzzy system is described in (Kim and Kosko, 1996) for predicting and filtering one-dimensional signals corrupted by impulsive noise. The paper opens with a review of the properties of noise having alpha stable statistics as a model for impulsive noise processes (Gaussian noise statistics is a special case of noise with alpha stable statistics). A new pseudo covariance measure is derived for such noise processes, which do not have finite second moments. One of the two fuzzy systems that are compared uses the 'Standard Additive Model' (SAM), which is essentially a Mamdani-type system with defuzzification carried out using only the centroids of the output fuzzy sets. The problem of rule base explosion with increase in dimensionality is pointed out and it is suggested that effective rule base identification is the answer to this problem. In the first fuzzy system, an adaptive vector quantisation with competitive learning method is used to identify ellipsoids in input-output space using training data. These ellipsoids are projected onto the input axes to form triangular fuzzy set member functions. The output fuzzy sets are formed by projecting the centroids of the learned ellipsoids onto the output axis. In this fuzzy system the conjunction operator AND is effected using one of the normal 'Min' or 'Product' operators. The second fuzzy system described differs in that there are no separate fuzzy sets for each component of the input vector. Instead, only a joint fuzzy set value is calculated for all the antecedents in each rule. This joint fuzzy set value is calculated using the 'Mahalanobis distance' of the input vector from the rule centroid. The Mahalanobis distance measure is better able to preserve the correlation information between the input data vector components than the usual factored form of antecedent. Two fuzzy filters based on the above ideas are demonstrated at the end of the paper and compared with a radial basis function filter network. The filters are applied to a polynomial signal corrupted with Cauchy noise and the best performance measure by mean squared error is obtained by the fuzzy filter using the Mahalanobis distance based conjunction operator. Table 6.1 is a summary of the fuzzy filters reviewed here

Paper Reference	Direct/ Indirect	Inferencing method	Input Variables	Training	Summary
(Arakawa and Arakawa, 1991)	Indirect	FAM type	Difference from centre value	Gradient descent of stepwise approximation	Adjusted weighted mean filter
(Kwan and Cai, 1993)	Indirect	FAM type	Centre, median, max and min values		Adjusted weighted Mean filter
(Taguchi *et al.*, 1994)	Indirect	Mamdani	Difference from centre value, distance from centre	Heuristics plus manually adjusted	Adjusted weighted Mean filter
(Takashima *et al.*, 1995(a))	Indirect	Zero order Sugeno	As (Taguchi *et al.*, 1994) & local variance	Back-propagation	Adjusted weighted mean filter
(Takashima *et al.*, 1995(b))	Indirect		Difference from reference & distance from centre	Heuristics plus manually adjusted	Adjusted weighted Mean. Two fuzzy stages
(Peng and Lucke, 1994)	Indirect	FAM type	Difference from centre value	Iterative gradient estimate	Adjusted weighted mean Data pre-filtered
(Peng and Lucke, 1995)	Indirect	FAM plus Mamdani for scaling stage	As [PENG94] & local variance for FAM scaling	Scaling stage training-heuristic	As (Peng and Lucke, 1994) & additional fuzzy scaling stage
(Hsiao and Lai 1995).	Indirect	FAM type	Residuals of Least Squares algorithm	Input MFs set up according to variance of residuals	Input data modified by fuzzy system. Ls then re-run
(Yang and Toh 1995)	Indirect	FAM	Difference from median within each level	Heuristic	MLMF adapted by fuzzy stage
(Russo, 1996)	Indirect	FIRE	Difference from centre value	Heuristics	Correction to centre value
(Arakawa, 1996)	Indirect	FAM	Difference of centre value, median & 2 closest values to centre.	FAM trained by gradient descent	Conditional median
(Mancuso *et al.*, 1996)	Indirect	Zero order Sugeno	difference from centre value	Heuristic	Conditional median
(Kim and Kosko,1996)	Direct	Mamdani (SAM)	Two sample differences	Competitive learning of input MFs	Direct acting Mamdani.

Table 6.1 Summary of review of fuzzy filters.

6.5 Indirect acting fuzzy logic-based filters (indirect FLBF)

6.5.1 Introduction

As outlined in section 6.4, indirect acting filters form by far the majority of fuzzy filter architectures described in the literature. Such filters have been applied to the task of greyscale image restoration. However, other than the work reported here and in (Rothwell Hughes *et al.*, 1995) and (Rothwell Hughes *et al.*, 1996), it is believed that there are no reports of their use to smooth depth or disparity maps. Because the fuzzy filters based on the FIR type Sugeno inferencing system described in section 6.6 represent a development of the indirect fuzzy filter architecture, the development of an indirect FLBF is described here. The performance of the filters based on the FIR type Sugeno system is examined in Chapter 7.

6.5.2 Description of indirect FLBF

The indirect acting fuzzy logic-based filters, which follow the adjusted weighted mean architecture, are structured in a similar way to one-dimensional or two-dimensional finite impulse response filters (figure 6.2). The filter weights applied to each pixel in an n x n filter window are determined by the output of a fuzzy logic system. The overall filter output is then given by equation 6.1. The general structure of such a filter is illustrated in figure 6.2.

Figure 6.2 depicts a two-dimensional filter, but the structure is essentially the same for a one-dimensional filter with the n x n filter window replaced with a 1 x n window. The pre-processor stage can be included to derive inputs to the fuzzy inferencing stage from the raw input data. If the filter design is such that only raw data is input to the fuzzy weight-determining stage, then the pre-processor stage is omitted. The firing weights, which multiply the input data in the window, are determined by the output of the fuzzy stage. The remainder of the filter implements equation 6.1.

A feature of this filter architecture is that the fuzzy inferencing system must either be a multiple-input, multiple-output (MIMO) system, or must be a multiple-input single-output (MISO) system applied n x n times for each output value. This point is illustrated in section 6.5.4 which discusses an indirect fuzzy filter structure that has been used to smooth dense disparity maps (*op cit* Rothwell Hughes *et al.*, 1995).

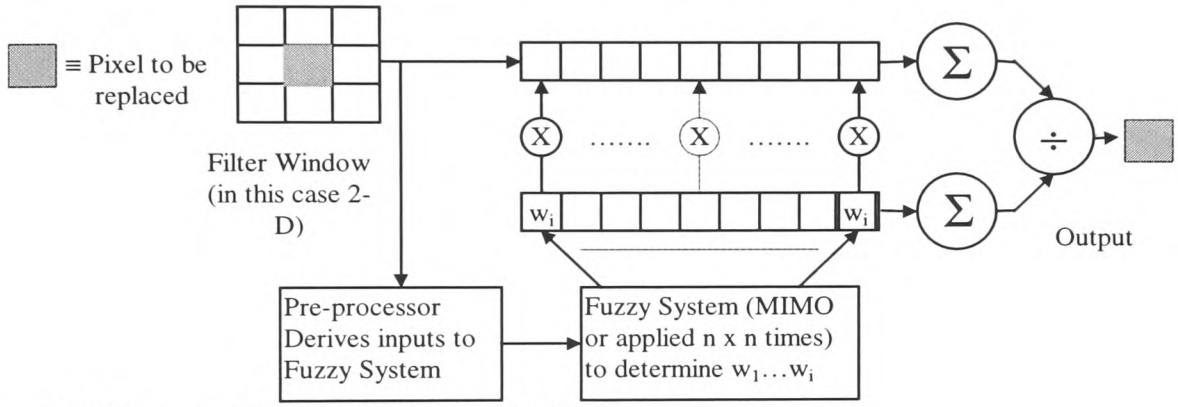

Figure 6.2 Structure of general indirect fuzzy filter

6.5.3 Difference from median pre-processor

As can be seen from table 6.1 a majority of the papers listed describe fuzzy filters with the architecture of figure 6.2, in which the pre-processor stage extracts the difference from some reference value for each input value in the filter window. The choice of this reference value is made so that a high difference indicates that the input value concerned does not belong to the same group or class as the centre value that is being filtered. The degree to which the input value is related to the centre pixel is determined by the fuzzy inferencing system, which then weights the input value accordingly. If the centre input value itself is chosen as the reference then the filter can be made to behave as a weighted mean or transversal filter, which does not smooth over abrupt

boundaries in the signal. This approach is only effective, however, when the signal to noise ratio at boundaries is high and there is a low probability of impulsive noise. In poor signal to noise regimes or when the noise is impulsive, the centre input value is quite likely to be corrupted itself and is thus an unreliable reference.

As discussed in section 4.6.1, the median value of an input filter window is stable in the presence of impulsive noise and preserves edge structures. The use of the median as a reference in an indirect filter offers two advantages over the centre value. Firstly, the difference from median (DFM) is a robust predictor of whether a pixel belongs to one side or another of an edge. Secondly, the difference from median is robust to impulsive noise. The DFM therefore serves as a good measure of outliers due to signal discontinuities and impulsive noise. A fuzzy system can take this measure of the degree to which an input signal value is an outlier and apply a soft thresholding operation. This soft thresholding operation applies a low weight to input signal values with a high possibility of being outliers. Once the outliers have been attenuated by being given a low weight the probability distribution of the resulting sample should be shorter tailed and the sample should be taken only from homogenous signal areas. Thus after applying the weighting, the signal has been conditioned by being segmented into homogenous areas, and the impulsive noise component should be removed. As noted in (Pittas and Venetsanopoulos, 1990) the arithmetic mean is a better (in an asymptotic relative efficiency sense) filtering strategy than the median for data that have distributions with shorter tails than the Laplacian distribution. Thus after the weighting operation is carried out the arithmetic mean of the weighted values is an appropriate strategy, providing the weighted sample has a shorter tailed distribution than the Laplacian.

In order to demonstrate the ability of the DFM measure to detect noise impulses, a hard logic filtering system was implemented. The hard logic filter was governed by the rule:

IF DFM is > threshold THEN filter weight assigned to value is Zero. R6.1

The result of applying this hard logic filter to a signal corrupted by mixed Gaussian and impulsive noise is shown in figures 6.3 to 6.8. Figure 6.3 shows a constant signal corrupted by mixed impulsive and Gaussian noise, with the impulsive noise component showed in figure 6.4. Figure 6.5 shows the DFM signal where the median is evaluated over a 1 x 5 window. Figure 6.6 is a hard thresholded version of the DFM signal, which represents detected impulses. This hard thresholding corresponds to a crisp rule based system. Figure 6.7 shows the distribution of the noise-corrupted signal. A Gaussian curve of the same mean and variance as the Gaussian component of the mixed noise is superimposed on the histogram to show the long tailed nature of the signal distribution. Figure 6.8 shows the distribution of the signal after removal of the impulses detected by the difference from median signal. The resulting histogram has fewer samples in the tail of the distribution than before the impulses were removed, although it still contains some outliers.

The effect of the same hard logic filter system on a step signal contaminated by Gaussian noise is illustrated in figures 6.9 to 6.17. Figure 6.9 represents an ideal step signal and the passage of filter windows over the step. If this signal is contaminated with Gaussian noise, then the distribution of values in the filter windows h2 to h5 over a large number of such steps will be bi-modal. An averaging filter is clearly not ideal for these bi-modal distributions. If segmentation could take place such that the values in the diagonally striped parts of the filter windows were excluded, then the residual values could be filtered with an averaging filter.

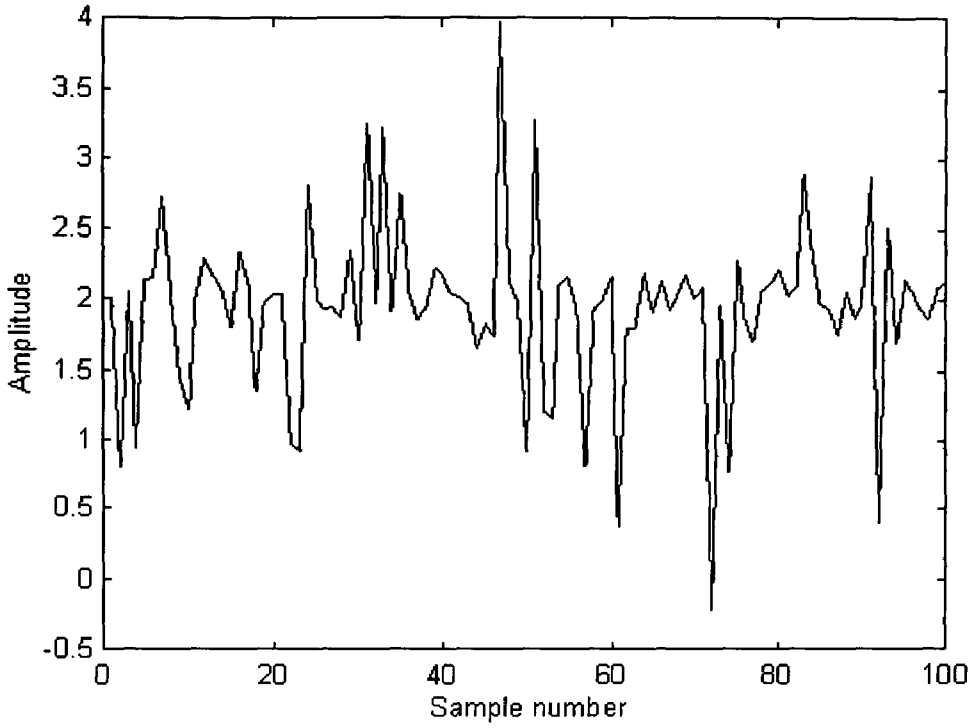

Figure 6.3: Constant signal corrupted by mixed impulsive and Gaussian noise

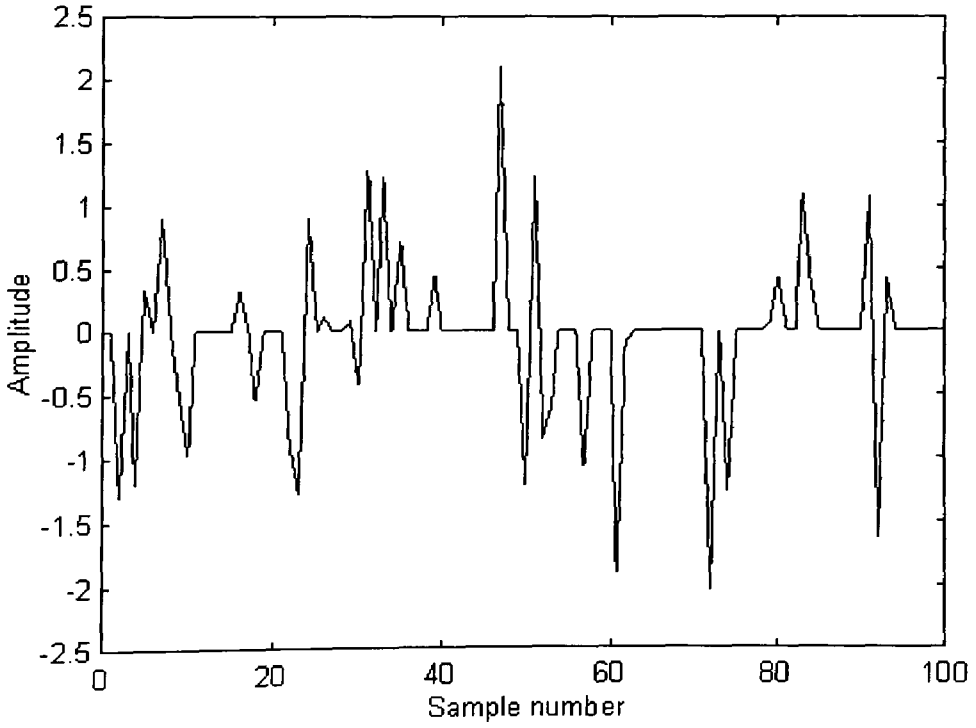

Figure 6.4: Impulsive noise component of signal shown in figure 6.3

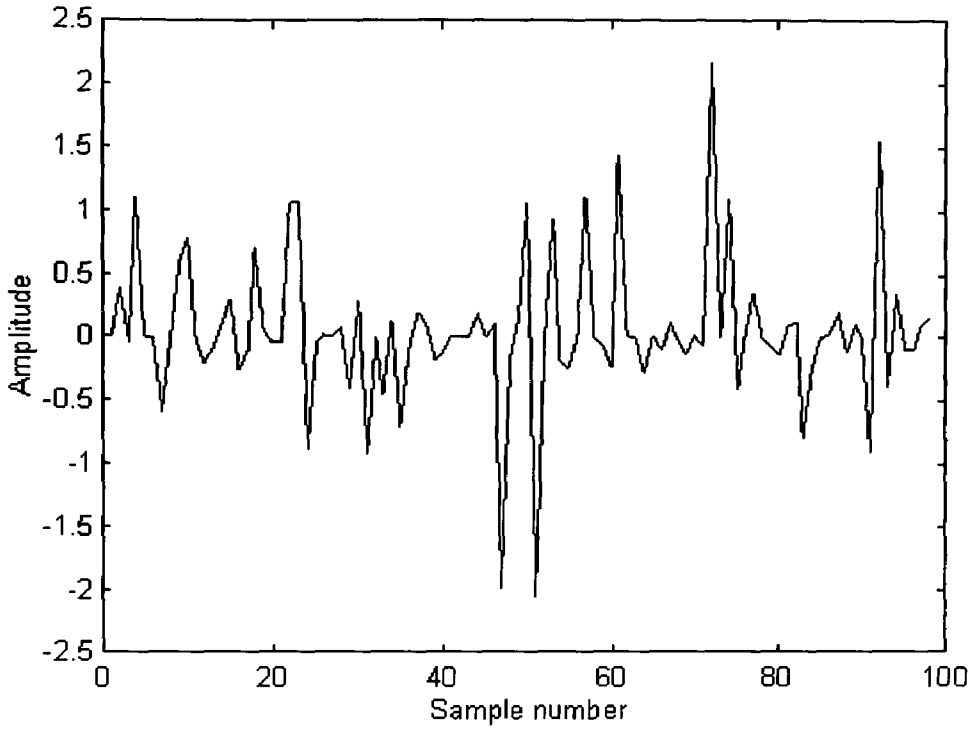

Figure 6.5: DFM signal

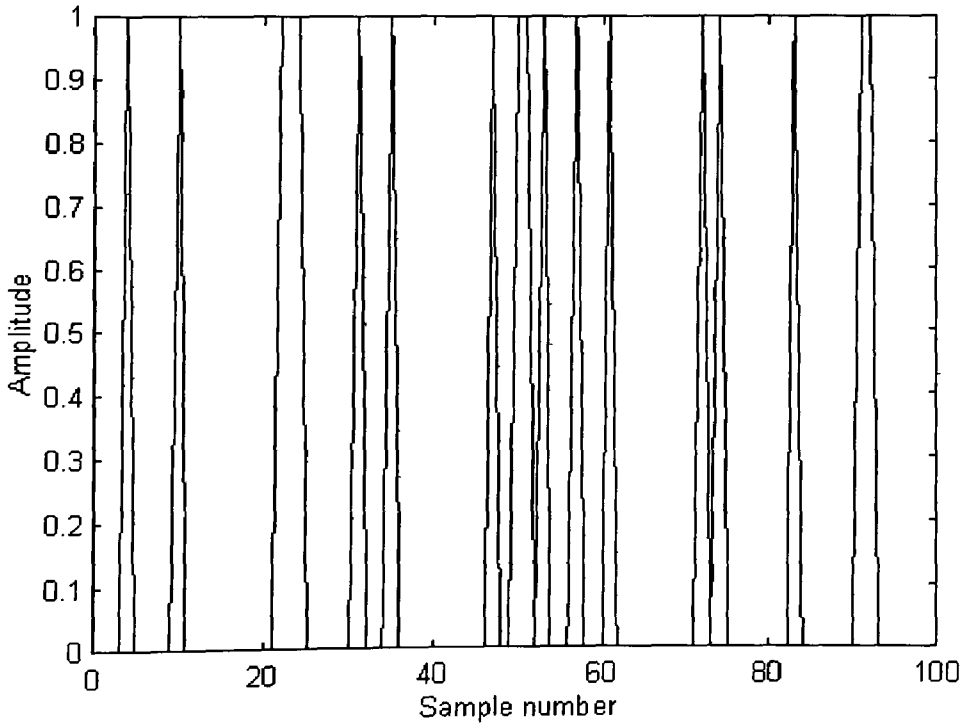

Figure 6.6: Hard thresholded version of the DFM signal, which represents detected impulses

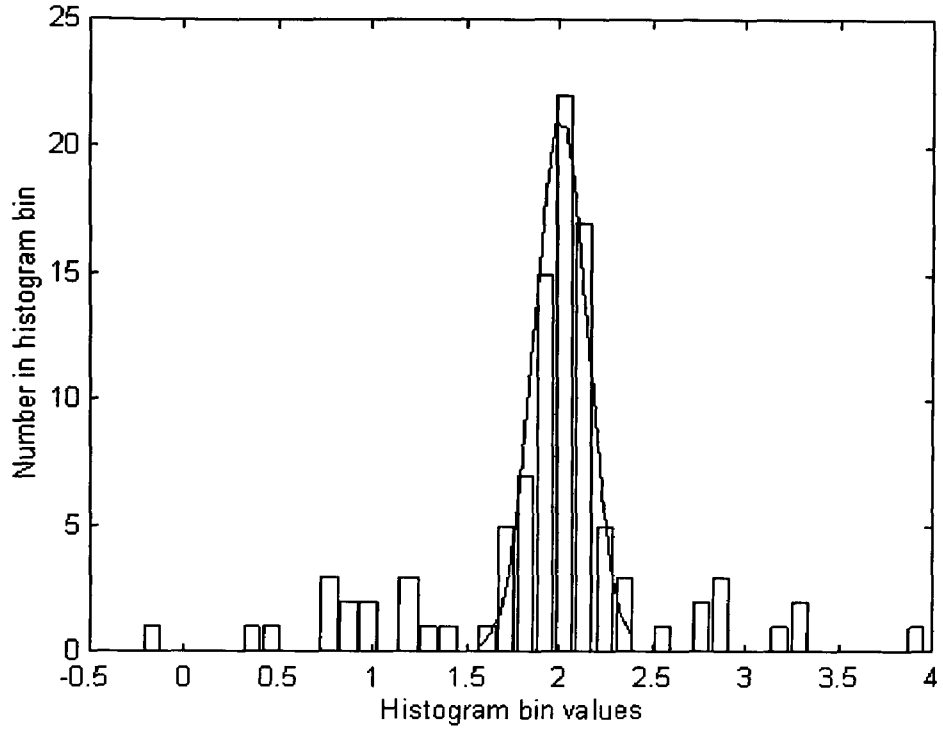

Figure 6.7 Histogram of the noise-corrupted signal

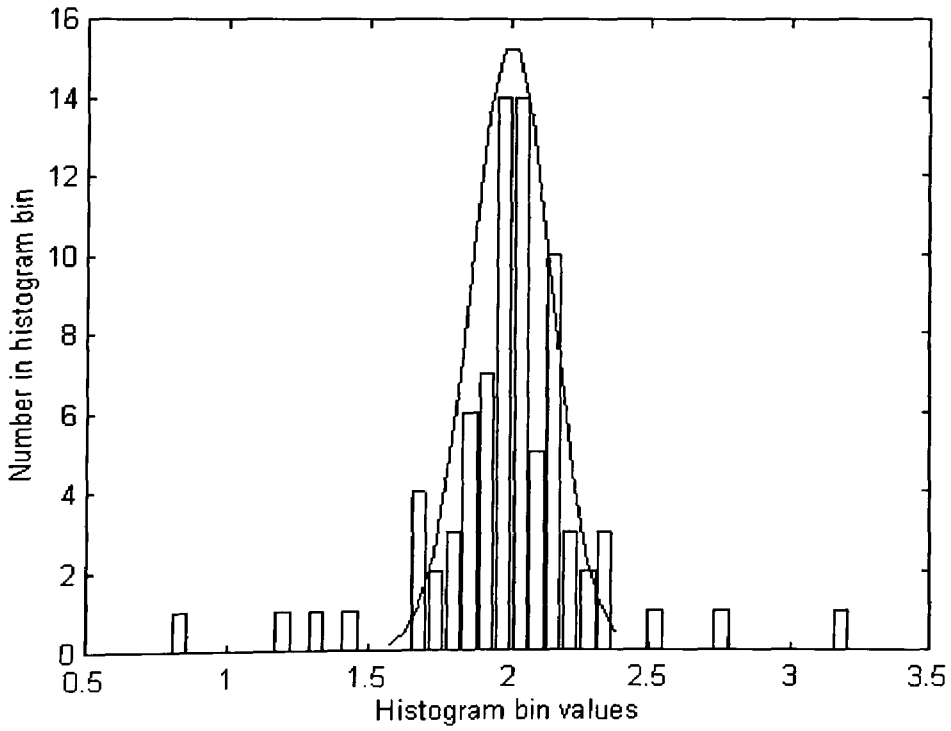

Figure 6.8: Histogram of signal after removal of detected impulses

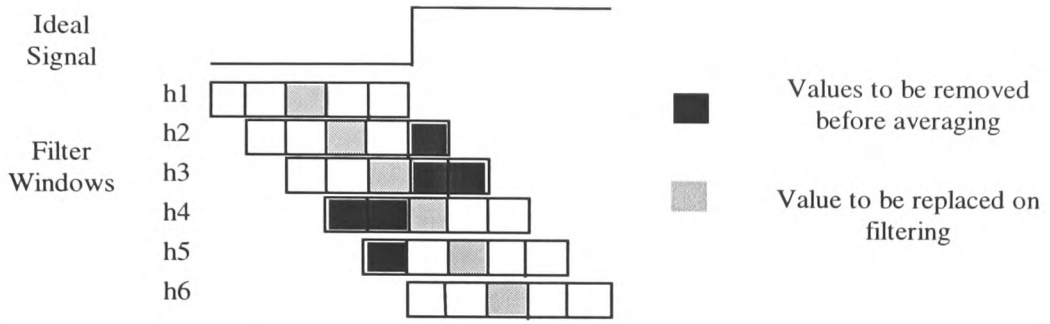

Figure 6.9: Representation of an ideal step signal and the passage of filter windows over the step

By applying the same rule, R6.1, the DFM measure can achieve the required segmentation. This is illustrated in figures 6.10 to 6.17 which show the filter window histograms before and after applying the rule R6.1 to a step signal which changes value from 3 to 4 and is corrupted by Gaussian noise of variance 0.09. The spike in the histograms after segmentation at zero represent values which have been rejected by the rule R6.1, i.e. they would be given a weighting of zero in the subsequent weighted average process. The histograms are unimodal after applying the rule R6.1, giving a resultant signal for which averaging is an appropriate filtering strategy.

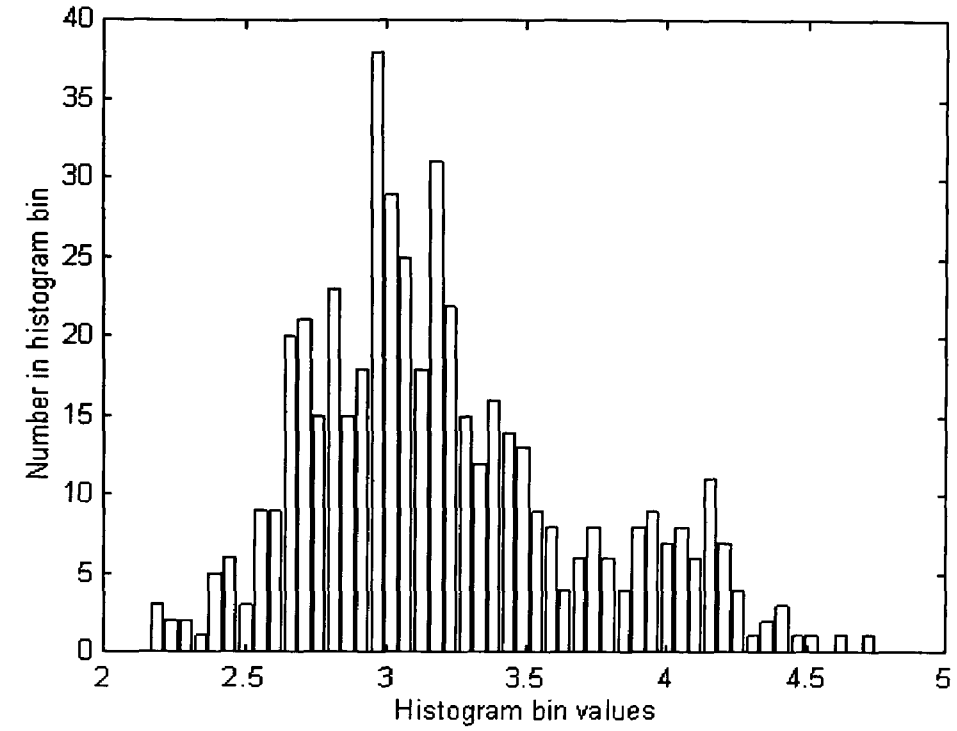

Figure 6.10: Histogram of filter window h2 values before segmentation

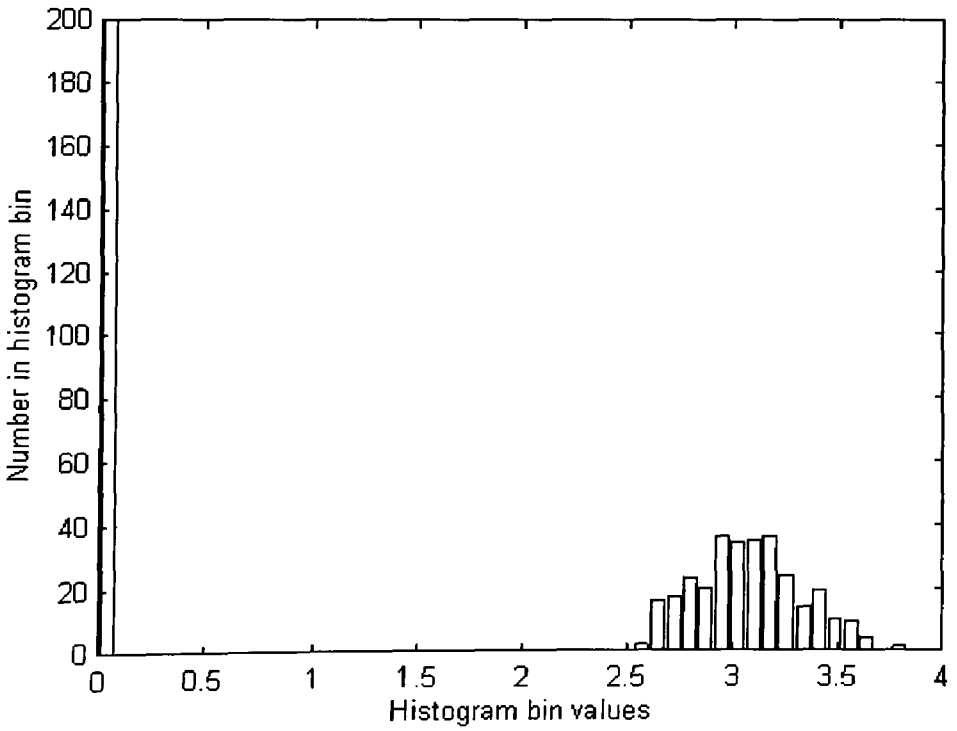

Figure 6.11: Histogram of filter window h2 values after segmentation

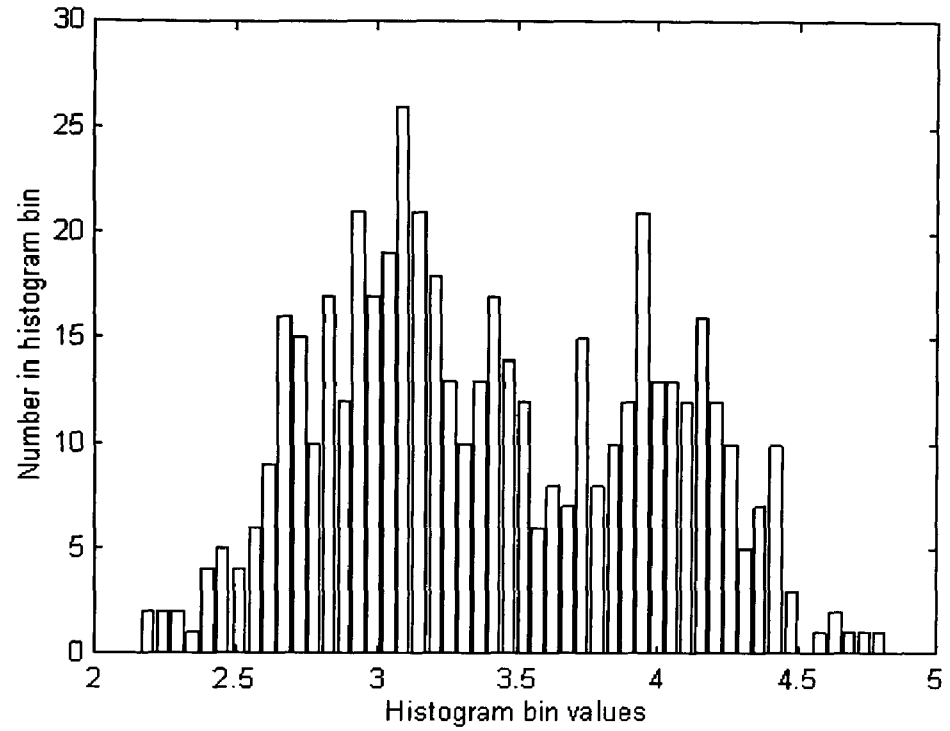

Figure 6.12: Histogram of filter window h3 values before segmentation

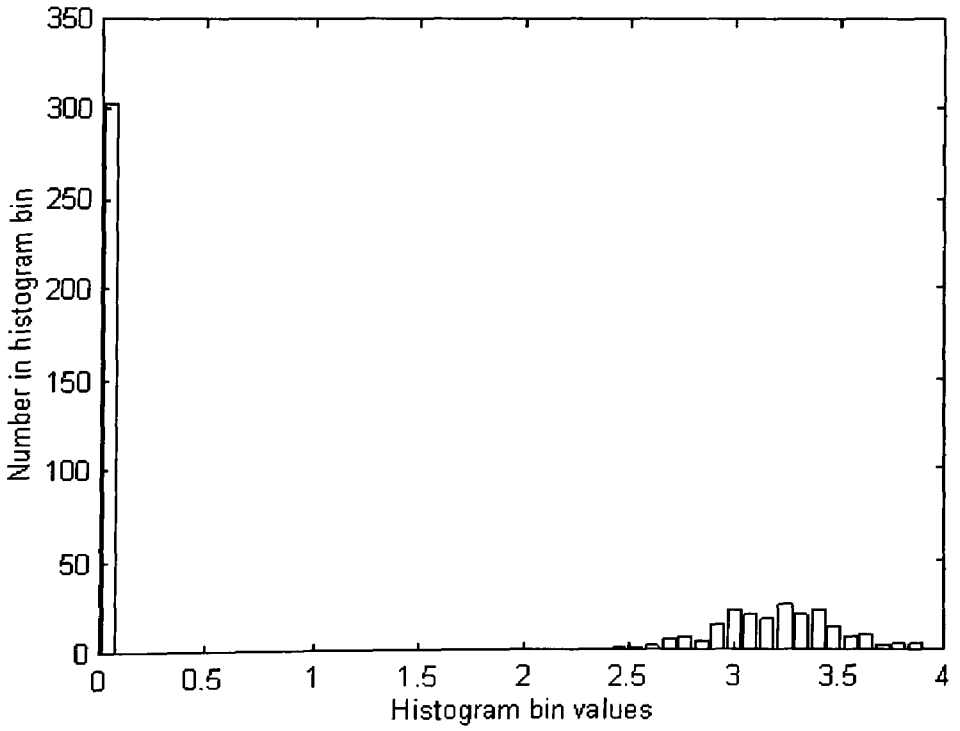

Figure 6.13: Histogram of filter window h3 values after segmentation.

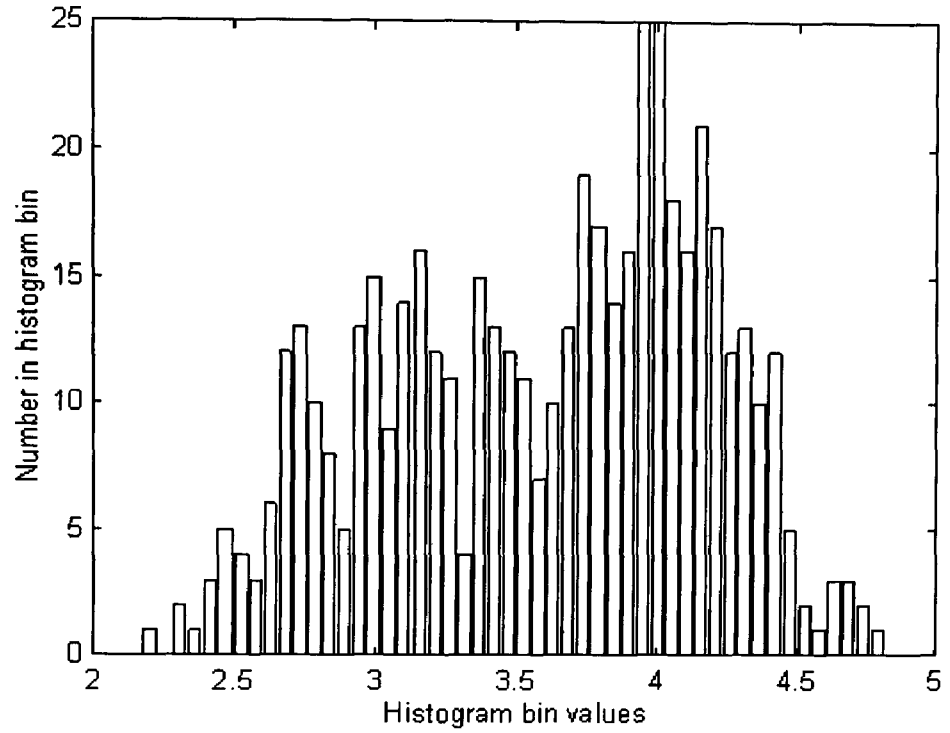

Figure 6.14: Histogram of filter window h4 values before segmentation

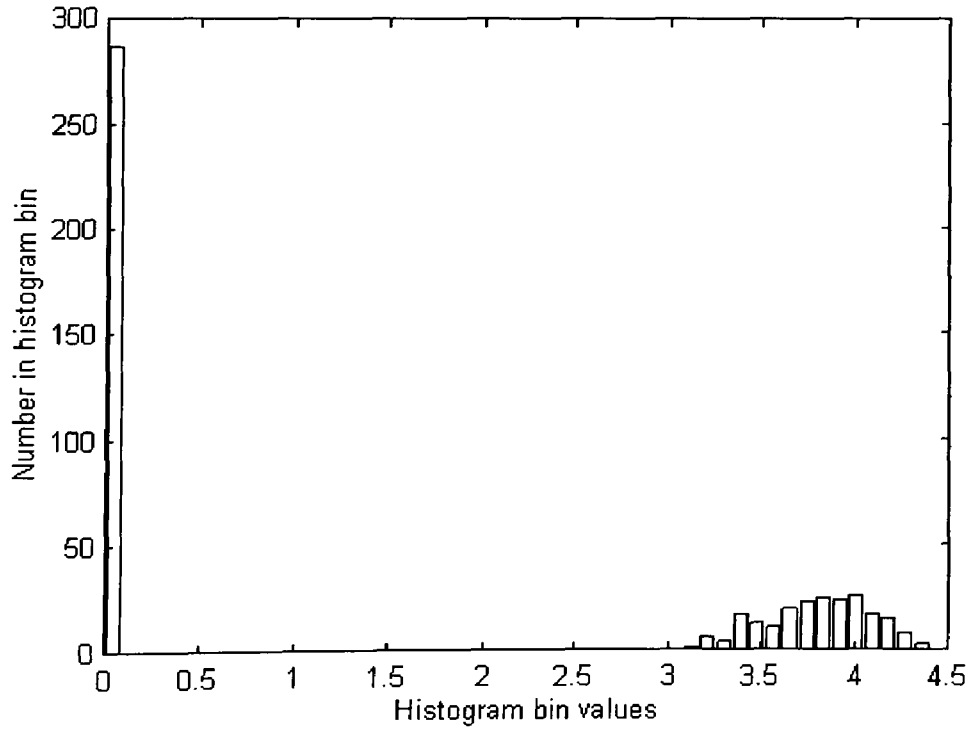

Figure 6.15: Histogram of filter window h4 values after segmentation

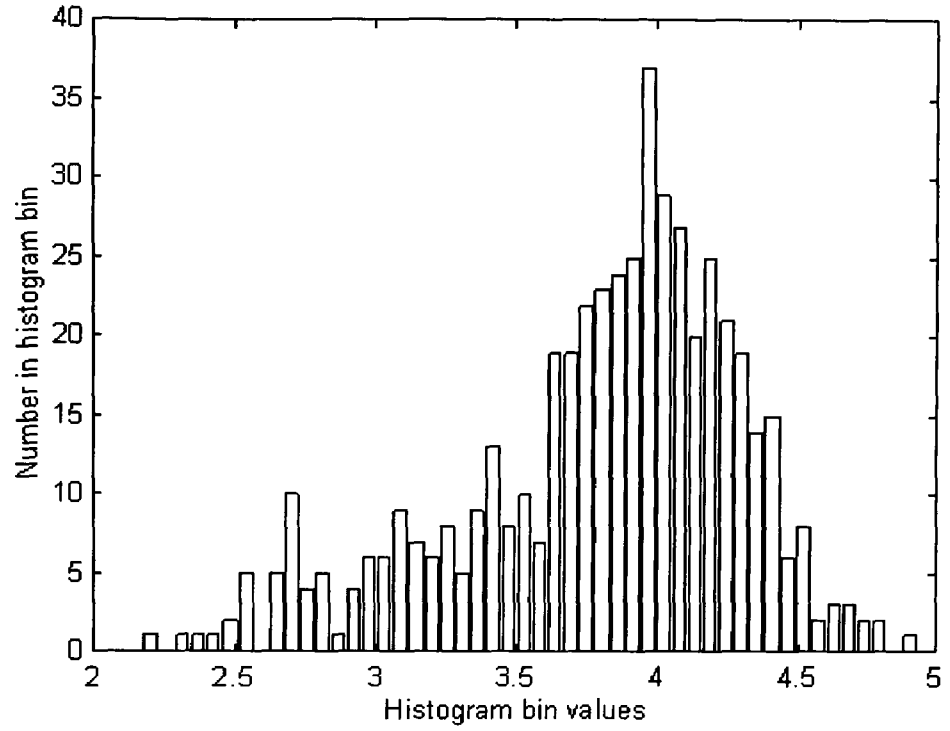

Figure 6.16: Histogram of filter window h5 values before segmentation

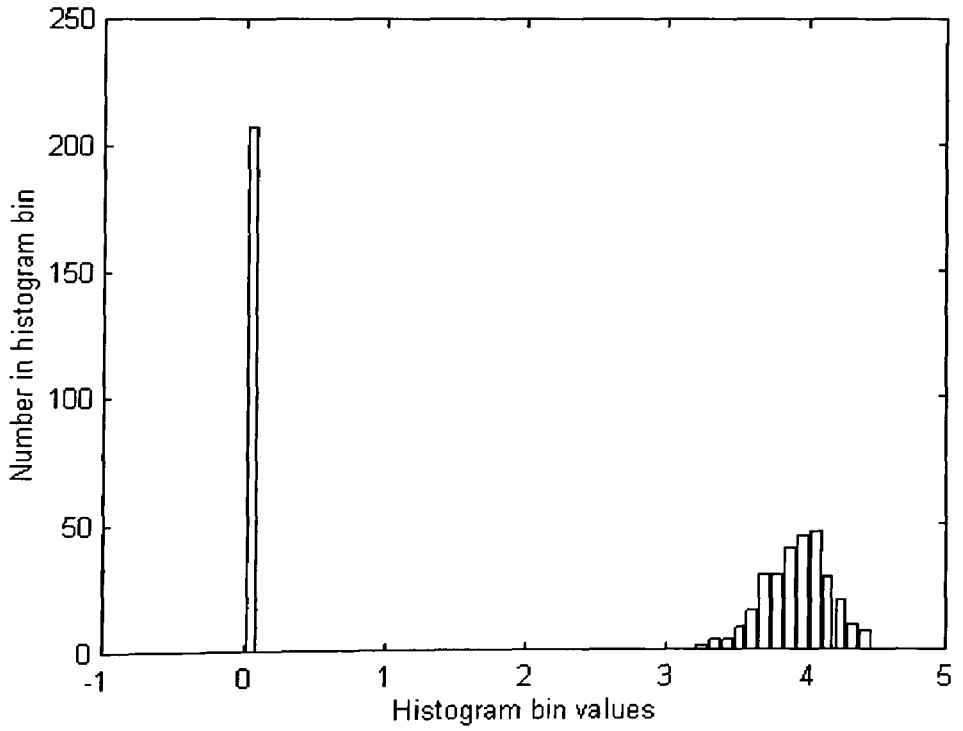

Figure 6.17: Histogram of filter window h5 values after segmentation

6.5.4 An Indirect Fuzzy Filter with Difference from Median Pre-processor.

In this section, an indirect fuzzy filter is described which is designed to smooth disparity maps derived from sequences of images as described in section 2.3.4. These disparity maps are typically corrupted by impulsive noise. The hard logic approach described in section 6.5.3 is extended by using a fuzzy threshold. An example of the use of this filter on disparity maps derived from real images is shown in Section 6.5.5.

In the hard logic system a pixel value is either accepted or rejected within a filter window, depending on a hard threshold applied to the difference from median. This is equivalent to classifying a pixel as being of the same class as the majority of other pixels in a window or not. The fuzzy approach allows a pixel to have some degree of membership in this class. The structure of the fuzzy filter has the architecture of figure 6.2 with a difference from median pre-processor and a 5 x 5 filter window. In addition, the distance of the pixel being weighted from the centre pixel of the filter window is taken as an input to the fuzzy weigh-determining stage. The crisp input values that are fuzzified (the observation variables) are:

The Euclidean distance of a pixel whose weight is being evaluated, Pixel($i+h,j+k$) from the pixel to which the smoothed disparity output will be assigned, Pixel(i,j): This distance is:

$$pixel\ dist = \sqrt{h^2 + k^2} \tag{6.4}$$

The difference in disparity of pixel ($i+h,j+k$) from the median of the disparities in the 5 x 5 filter window, d_{median} (the DFM) normalised to the value of the median. This disparity difference measure is:

$$disparity\_difference = \left| \frac{d(i+h, j+k) - d_{median}}{divisor} \right| \qquad 6.5$$

where $divisor = d_{median}$ when $d_{median} > 1$, and $divisor = 1$ otherwise.

The disparity measure is scaled to the local median disparity, because in this application the disparity will increase through the sequence of images as the camera moves. By normalising to the median, the universe of discourse for the *disparity_difference* fuzzy sets does not have to cover the full range of disparities that might be encountered. The need for this normalisation illustrates a potential problem with filters based on fuzzy logic, which is that they can only implement a desired mapping over a fixed and pre-defined universe of discourse.

The three triangular fuzzy input sets for pixel distance are labelled *Near*, *Local,* and *Far*. They have membership functions as shown in figure 6.18

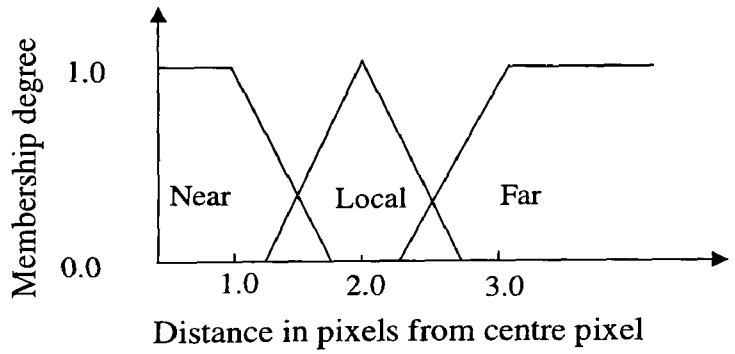

Figure 6.18: Pixel distance membership functions for indirect filter of section 6.5.4.

The three sets for *disparity_difference* are labelled *Small*, *Medium*, and *Large* with triangular/trapezoidal membership functions and are shown in figure 6.19. The shapes of the membership functions are chosen to be triangular/trapezoidal for ease of computation. They need not in general be symmetrical. The fuzzy rule base and inference engine combine the antecedent fuzzy sets two at a time and associate them with an output or consequent set. The fuzzy rule base

used is shown in table 6.2. The output or consequent sets are fuzzy singletons corresponding to a zero order Sugeno system. The membership functions for the output sets are shown in figure 6.20 and they are labelled O_i, i=1,2,..5.

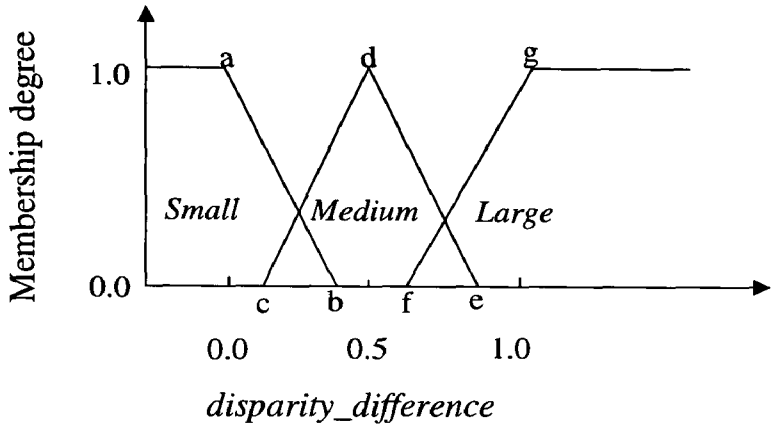

Figure 6.19: Input sets for *disparity_difference* for indirect filter of section 6.5.4.

Consider the following disparity values in a 5 x 5 filter window :

4 3 4 5 3

2 4 2 3 2

4 7 [5] 4 5 The window median = 4

4 5 6 5 4

5 4 4 4 3

The centre disparity value in the box is the pixel to be filtered. The underlined disparity is at a distance 1 pixel from the centre pixel, and so its membership values in the distance sets are:

$$\{M_{near}(1), M_{local}(1)M_{far}(1)\} = \{1 ,0 ,0 \} \quad 6.6$$

The *disparity_difference* is (7-4)/4 = 0.75 with membership values:

$$\{M_{small}(1), M_{medium}(1)M_{large}(1)\} = \{0.0 ,0.33 ,0.33 \} \quad 6.7$$

Pixel_distance	disparity_difference		
	Small	Medium	Large
Near	O_5	O_3	O_2
Local	O_4	O_3	O_1
Far	O_3	O_2	O_1

Table 6.2: Rulebase for indirect filter of section 6.5.4.

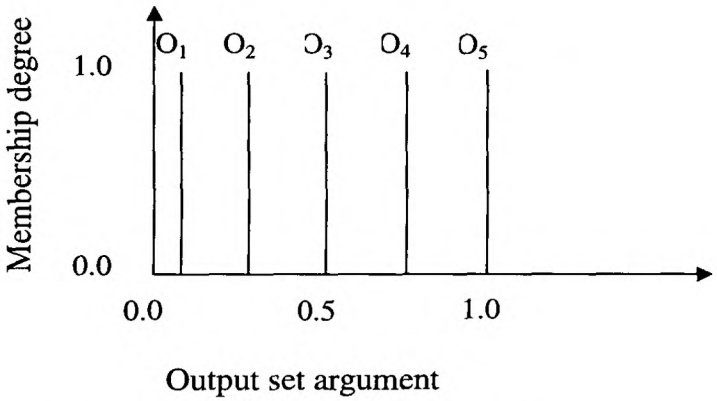

Figure 6.20: Singleton output sets for indirect filter of section 6.5.4.

The rule base reflects the *a priori* assumptions about the correct form of the disparity map; pixels are correlated in proportion to their separation and pixels of markedly different disparity belong to different surfaces. The first rule in the rule base can be read as:

IF *pixel_distance* is Near AND *disparity_difference* is Small Then Output is O_5 R6.2

For this rule, a weight w_1 is assigned to the consequent fuzzy singleton O_5. The weight is determined by taking the minimum of the antecedent membership function values. i.e.

$$w_1 = \min[\, M_{near}(1),\, M_{small}(0.75)] = \min[1,0\,] = 0 \qquad 6.8$$

This is repeated for all the rules in the rule base, until a weight w_i is associated with each rule R_i and its consequent fuzzy singleton O_i. The final crisp output, $O(h,k)$ is computed by the weighted sum over all the N rules:

$$O(h,k) = \sum_{i=1}^{N} \frac{w_i O_i}{w_i} \tag{6.9}$$

This crisp output is now taken as the weight of pixel $(i+h, j+k)$. The process of obtaining filter weights is repeated for all the pixels in the 5 x 5 filter window. The final filtered output which replaces the unfiltered value of disparity for pixel (i ,j) is calculated from:

$$d\,fil(i,j) = \frac{\sum_{h=-2}^{h=+2} \sum_{k=-2}^{k=+2} O(h,k) d(i+h, j+k))}{\sum_{h=-2}^{h=+2} \sum_{k=-2}^{k=+2} O(h,k)} \tag{6.10}$$

6.5.5 Example: Application of the Indirect Fuzzy Filter to a noisy disparity map

The parameters of the indirect fuzzy filter described in section 6.5.4 were adjusted manually on the disparity map resulting from a random dot simulated image sequence for which the disparity map is known exactly. The resulting observation variable to filter weight mapping is illustrated in figure 6.21. Figure 6.22 shows the first image in an image sequence 'Toys'. This sequence of images was taken of a static scene under known camera motion. Using the approach described in section 2.3 and the SSD matcher discussed in Chapter 3, a noisy disparity map was generated as shown in figure 6.23. Figure 6.24 shows the disparity map generated for the image sequence 'Toys' after the manually tuned indirect acting fuzzy filter was applied between each matching step.

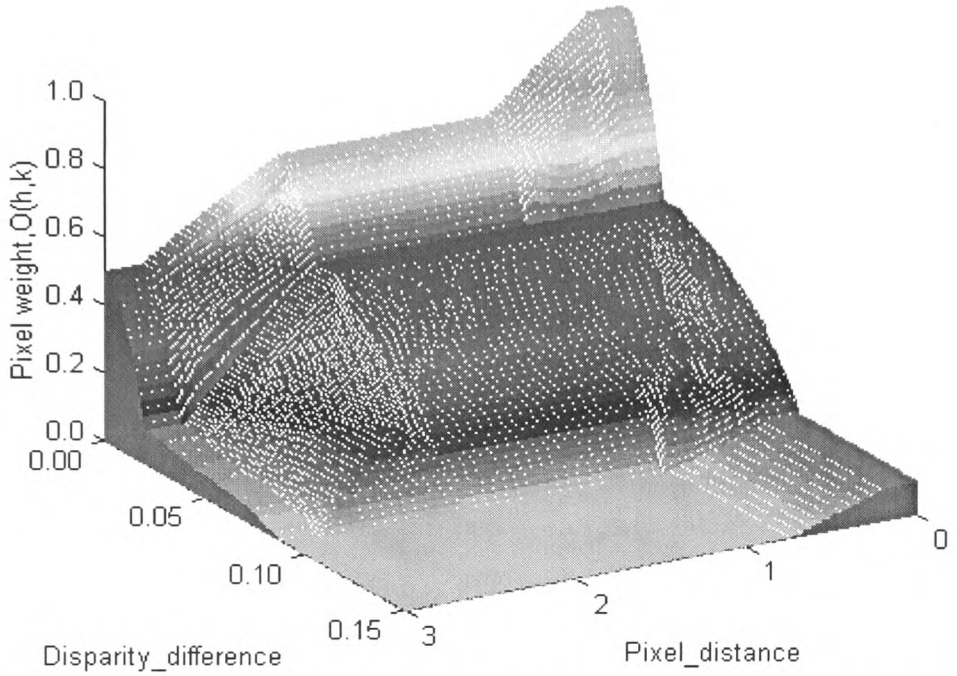

Figure 6.21: Observation variable to filter weight mapping for indirect fuzzy filter of section 6.5.5

Figure 6.22: First image in image sequence 'Toys'

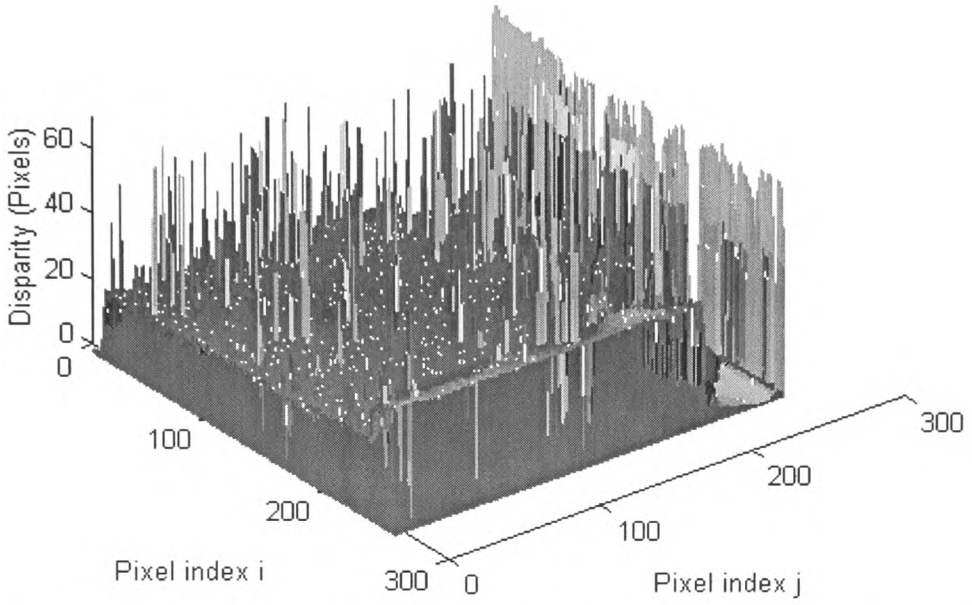

Figure 6.23: Noisy disparity map generated from image sequence 'Toys'

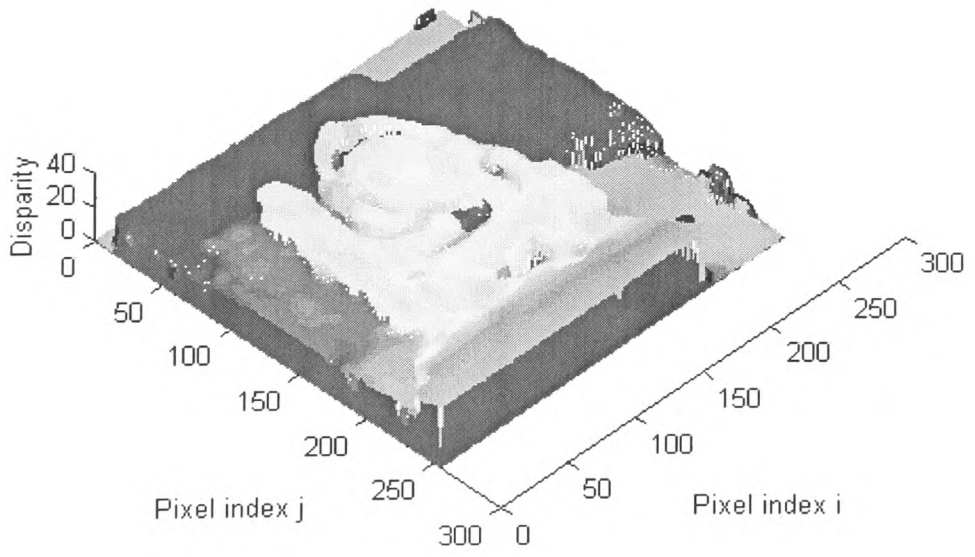

Figure 6.24: Disparity map generated for the image sequence 'Toys' after the manually tuned indirect acting fuzzy filter was applied between each matching step.

6.6 A direct acting FLBF using the Sugeno inferencing system

6.6.1 Architecture of the Sugeno FIR type FLBF.

The structure of the Sugeno FIR type FLBF is based on the FIR type Sugeno fuzzy system described in section 5.3.2 in which each output set is defined by its coefficient set, a_i. If the coefficients a_i are chosen appropriately, each output set can be regarded as a transversal or finite impulse response (FIR) filter. Thus, the FIR type Sugeno system can be drawn as a feedforward nonlinear filter structure as shown in figure 6.21.

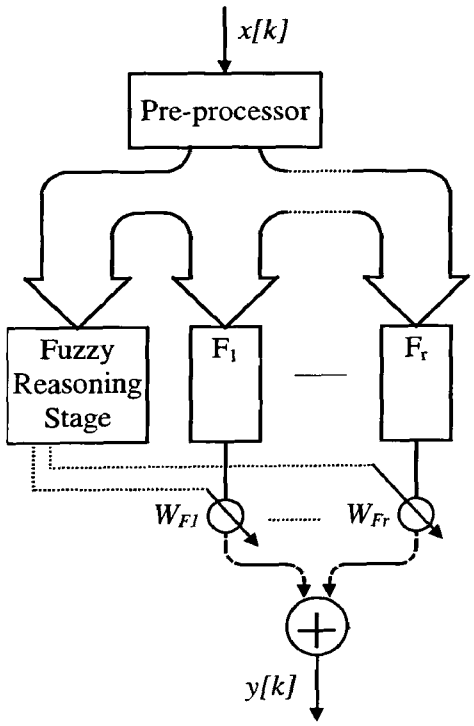

Figure 6.25: FIR type Sugeno system drawn as a feedforward filter structure

The pre-processor assembles the raw depth data samples from the filter window into a correctly ordered vector for input to the FIR type Sugeno system. The blocks F_1 to F_r represent the output

sets of the first order Sugeno network. There are as many of these blocks as there are output sets in the Sugeno network. The weights W_{F1} to W_{Fr}, which are applied to the outputs of the linear filters are the firing weights of each rule for the applied inputs $x[k]$. The final output of the filter is the weighted sum of the outputs of the filter bank with the weights determined by the fuzzy stage as a nonlinear function of the pre- processed input. Viewed in this way it can be seen that the FIR type Sugeno network can implement a general nonlinear filter. The structure can also implement any linear FIR filter as a special case by fixing the output of the fuzzy inferencing stage to select the appropriate filter or combination of filters regardless of the input. However, the advantage of this filter structure lies in its ability to choose the output coefficient set, and hence the filter action, depending on the inputs to the filter. This leads to an overall nonlinear filter, which in effect interpolates multiple linear filters. In control applications this is described as a gain scheduling approach (Jang and Gulley, 1994), but it is believed that the explicit identification of this approach with the idea of a filter structure is a novel idea. The use of the FIR type Sugeno filter offers the potential advantage over previously proposed fuzzy filters that the choice of the coefficients for the different output sets can be guided by conventional filter design techniques. It is also easier to train the filter output coefficients and the input sets' membership functions using simulated input-output data with the FIR type Sugeno fuzzy filter structure than with the indirect FLBF.

6.6.2 Mapping performed by FIR type Sugeno fuzzy filter

Let there be N inputs defined for the fuzzy system, and let there be R rules. Let the pre-processor assemble the input data into the 1xN input vector $\boldsymbol{X}$, and let the feedforward fuzzy stage be represented by the function $f(.)$ which maps the 1 x N vector $\boldsymbol{X}$ into the 1xR rule firing-weight vector $\boldsymbol{W}$. Assume without loss of generality that each rule has a corresponding filter in the filter

bank. If the coefficients of the R filters in the filter bank are represented by the RxN matrix ***A***, the action of the filter can be described by the equation:

$$y[k] = \boldsymbol{W.A.X}^{\mathsf{T}} = f(\boldsymbol{X}).\boldsymbol{A.X}^{\mathsf{T}} \qquad 6.11$$

The function $f(.)$ is dependent on the choice of fuzzy rules and input fuzzy sets. If the rth fuzzy rule in a rulebase of R rules is :

$$IF\ x_1\ is\ FS_1^r\ AND \ldots . x_N\ is\ FS_N^r\ THEN\ W_r\ is\ w^r \qquad R6.3$$

Where the x_n are elements of the crisp input vector $\boldsymbol{X}=(x_1 \ldots x_N)$, the W_r are elements of the weight vector $\boldsymbol{W}=(W_1 \ldots W_r \ldots W_R)$, and the FS_n^r are fuzzy sets defined by Gaussian membership functions, then the rth element in the weight vector is given by:

$$W_r = \frac{min\{\mu_{FS1}^r(x_1), \ldots . \mu_{FSN}^r(x_N)\}}{\sum_{r=1}^{r=R} min\{\mu_{FS1}^r(x_1), \ldots . \mu_{FSN}^r(x_N)\}} \qquad 6.12$$

Where $\mu_{FSn}^r(x_n)$ is the membership of x_n in the fuzzy set FS_n^r.

The nonlinear feedforward mapping, $f(.): \boldsymbol{X} \rightarrow \boldsymbol{W}$ is determined by the rulebase and equation 6.12.

In order for the filter defined by the mapping of equation 6.11 to be useful in depth map regularisation, appropriate rules and input fuzzy sets have to be identified. However, for a 3 x 3 two-dimensional filter with two fuzzy sets per input, an exhaustive set of rules requires 2^9 rules. As discussed in section 5.11.1, this large increase in the size of the rulebase with filter size not only

makes the fuzzy system slow to evaluate the output for a given input, but also very hard to train effectively. Therefore, practical filters have to be generated using only a small subset of all the possible rules. A number of different approaches to using the FIR type Sugeno-based filters on one and two-dimensional signals are described in Chapter 7.

6.7 Summary of Chapter 6.

This chapter has presented a review of existing work on filters based on fuzzy logic and proposed a taxonomy for fuzzy filters based on the idea of direct and indirect acting filters. By far the majority of existing work on fuzzy filters describes indirect acting filters. Fuzzy-based filters of this type have previously been used to remove impulsive and mixed noise from greyscale images. An indirect acting fuzzy filter suitable for disparity map smoothing has been described in this chapter and used to smooth a noisy disparity map derived from a sequence of real images. The chapter concludes with a description of a direct acting fuzzy filter structure that is based on the FIR type Sugeno system. It is believed that the explicit identification of the FIR type Sugeno system as a filter structure represents a contribution made by this thesis.

6.8 References

(Arakawa and Arakawa, 1991) Arakawa K. and Arakawa Y. "Digital Signal Processing Using Fuzzy Clustering." IEICE Transactions Vol E74 No 11 November 1991.

(Arakawa, 1996) Arakawa K. "Median Filter based on Fuzzy rules and its application to image restoration." Fuzzy Sets and Systems Vol 77 No 1 pp 3-13 January 1996.

(de Oliveira, 1996). de Oliveira J. V. "Sampling, fuzzy discretization, and signal reconstruction." Fuzzy Sets and Systems Vol 79 pp 151-161 1996.

(Hornik *et al.*., 1989) Hornik K., Stinchcombe M., and White H. "Multilayer Feedforward networks are universal approximators." Neural Networks Vol. 2 pp 359-366, 1989.

(Hsiao and Lai 1995) Hsiao, Ch-H. and Lai, Ch-Ch. "Analysis and Design of Fuzzy Filter Algorithms" International IEEE/IAS Conference on Industrial Automation and Control: Emerging Technologies pp 413-420 1995.

(Jang and Gulley, 1994) Jang, J. S Roger and Gulley, N. "Gain Scheduling Based Fuzzy Controller Design." Proceedings of the International Joint Conference of the North American Fuzzy Information Processing Society Biannual Conference, The Industrial Fuzzy Control and Intelligent Systems Conference and NASA Joint Technical Workshop on Neural Networks and Fuzzy Logic. pp 101-105 1994

(Kim and Kosko, 1996) Kim H M and Kosko B "Fuzzy Prediction and filtering in impulsive noise" Fuzzy Sets and Systems Vol 77 pp15 -33. 1996.

(Kosko, 1992) Kosko B. "Fuzzy Systems As Universal Approximators" Proceedings of the IEEE International Conference on Fuzzy Systems, San Diego California, pp 1153-1162 March 1992.

(Kosko, 1992) Kosko B. "Neural Networks and Fuzzy Systems." Prentice-Hall 1992.

(Lee and Kassam, 1985) Lee Y.H. and Kassam S. A. "Generalised median filtering and related nonlinear filtering techniques" IEEE Transactions on Acoustics, Speech, and Signal Processing. Vol 33 No 3 pp 672-683 June 1985.

(Lee, 1980) Lee J. "Digital Enhancement and noise filtering by use of local statistics." IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol 2 No 2 pp 165-168, 1980.

(Mamdani, 1974) Mamdani, E. H. "Application of fuzzy algorithms for control of simple dynamic plant." Proceedings of the IEE- Control and Science; Vol 121 No12. pp 1585-1588. December. 1974.

(Mancuso *et al.*, 1996) Mancuso M., De Luca R., Poluzzi R., and Rizzotto G.G. "A fuzzy decision directed filter for impulsive noise reduction." Fuzzy Sets and Systems Vol 77 No 1 pp 111-116 January 1996.

(Peng and Lucke, 1994) Peng S, and Lucke L "Fuzzy Filtering for Mixed Noise Removal during Image Processing " Proceedings of the IEEE International Conference on Fuzzy Systems pp 89-93 June1994.

(Peng and Lucke, 1995) Peng S, and Lucke L "Multilevel Adaptive Fuzzy Filter for Mixed Noise Removal " Proceedings of the IEEE International Symposium on Circuits and Systems pp 1524-1527 1995.

(Pittas and Venetsanopoulos, 1990). Pittas I. and Venetsanopoulos A.N. "Nonlinear Digital Filters Principles and Applications." 1st Edition. Kluwer 1990.

(Pugmire *et al.* 1995) Pugmire R, Hodgson R, and Chaplin R. "The Properties and Training of a Neural Network Based Universal Window Filter (UWF)" Proceedings of Image Processing and its Applications IEE Conference. pp 642-646 July 1995.

(Pugmire *et al.* 1995) Pugmire R, Hodgson R, and Chaplin R. "The Properties and Training of a Neural Network Based Universal Window Filter (UWF)" Proceedings of Image Processing and its Applications IEE Conference. pp 642-646 July 1995.

(Rothwell Hughes *et al.*, 1995) Rothwell Hughes N., Wilson G. R., and Roberts G. N. "Kalman and Fuzzy Logic Filters for 3-D Industrial Inspection Using a Single Camera." Proceedings of the International Conference on Recent Advances in Mechatronics (ICRAM95). pp 595-600 August 1995.

(Rothwell Hughes *et al.*, 1996) Hughes N., Wilson G., and Roberts G., "Fuzzy Regularisation of Depth Maps Derived from Image Sequences", International Conference on Mechatronics and Machine Vision in Practice Vol 2 pp 55-60. 1996.

(Russo and Ramponi, 1994) "Nonlinear Fuzzy Operators for Image Processing" Signal Processing Vol 38 pp 429-440 1994.

(Russo, 1993) Russo F "A new class of fuzzy operators for image processing: Design and implementation." Proceedings of the IEEE International Conference on Fuzzy Systems pp 815-820 March1993.

(Russo, 1996) Russo F "Fuzzy Systems in Instrumentation: Fuzzy Signal Processing." IEEE Transactions on Instrumentation and Measurement Vol 45 No 2 pp 683-689 April1996.

(Taguchi *et al.*, 1994) Taguchi A, Takashima H and Murata Y "Fuzzy Filters for Image Smoothing." SPIE Conference on Nonlinear Image Processing V, Vol 1 pp332-339 1994.

(Takagi and Sugeno, 1983) Takagi T and Sugeno M, "Derivation of Fuzzy Control Rules from Human Operator's Control Actions." Proceedings IFAC Symposium on Fuzzy Information Knowledge Representation and Decision Analysis, pp 55-60. 1983.

(Takashima *et al.*, 1995(a)) Takashima, H.; Taguchi, A., and Murata, Y. "A Synthesis of an optimal fuzzy filter based on local statistics." Electronics and Communications in Japan Part III Fundamental Electronic Science. Vol 78 No 8 pp10-21 1995.

(Takashima *et al.*, 1995(b)) Takashima, H.; Taguchi, A., and Murata, Y. "Edge-preserving smoothing using the fuzzy control technique." Electronics and Communications in Japan Part III Fundamental Electronic Science Vol 78 No1pp72-43 1995.

(Wang and Mendel, 1992) Wang L.X. and Mendel J.M. "Fuzzy Basis Functions, Universal Approximation, and orthogonal Least-Squares Learning." IEEE Transactions on Neural Networks Vol 3 No 5, September 1992.

(Wang, 1992) Wang L.X. "Fuzzy Systems are Universal Approximators." Proceedings of the IEEE International Conference on Fuzzy Systems, San Diego California, pp 1163-1170. March 1992.

Chapter 7: Testing of Fuzzy Filters

7.1 Introduction

Chapter 6 has introduced a new architecture for fuzzy filters based on Sugeno inferencing systems that can be trained using the techniques described in Chapter 5. This chapter investigates the performance of the new fuzzy filter architecture when applied to the problem of filtering one-dimensional signals. It also applies filters based on the new fuzzy filter architecture to the task of filtering two-dimensional depth maps derived from real and simulated images.

The performance of the new fuzzy filter architecture is compared with that of moving average, Wiener, and median filters

7.2 WINIM software

This section is a brief description of the WINIM software that was written in order to investigate the application of fuzzy filters generated using the FTEST software that was described in section 5.3

The WINIM software is designed to read in pairs or sequences of images and carry out a stereo matching process as described in section 2.3 and the SSD matcher investigated in Chapter 3. In matching or sequence mode WINIM can carry out the following operations:

(a) Stereo matching can be performed on a pair of images taken with a known camera shift between images (under the assumptions (section 2.3.2) that the epipolar constraint holds

and that the camera is along the x-axis only). This step produces a disparity map and its associated variance map (section 3.5) and depth map.

(b) Filtering of the resulting depth map can be carried out using moving average, median, or any fuzzy filter generated using the FTEST software. The filter windows can be rectangular, square, or cross-shaped. The disparity map associated with the depth map is updated after filtering. Filtering is applied to the depth rather than the disparity map because the depth map has a constant range through a sequence of images, whereas a disparity map will evolve through the sequence.

(c) The steps (a) and (b) can be repeated with the next image in the sequence. WINIM also has the option of allowing the tracking of the disparity map using the Kalman filter described in section 2.3.4.

(d) The resulting depth map at each step is displayed as a greyscale image in the WINIM window. The depth map can also be saved for analysis and plotting in MATLAB.

(e) Simulated depth maps or other signals can also be read in from disk and filtered using the filtering options of WINIM and the resulting filtered signals saved to disk. Because of this last option, WINIM can be used to filter one-dimensional signals.

A screenshot of the WINIM main window is shown in figure 7.1.

7.3 Generation of simulated one-dimensional signal and noise

7.3.1 Introduction

The one-dimensional signals that are used as a source of test signals are generated using a program written in MATLAB. Another MATLAB program generates noise that is used to corrupt the one-dimensional signals. The signals are saved to disk from where they can be read by the WINIM image processing software introduced in section 5.3.1. The signals and

corrupting noise are designed to be one-dimensional equivalents of the depth maps and noise described in Chapter 2.

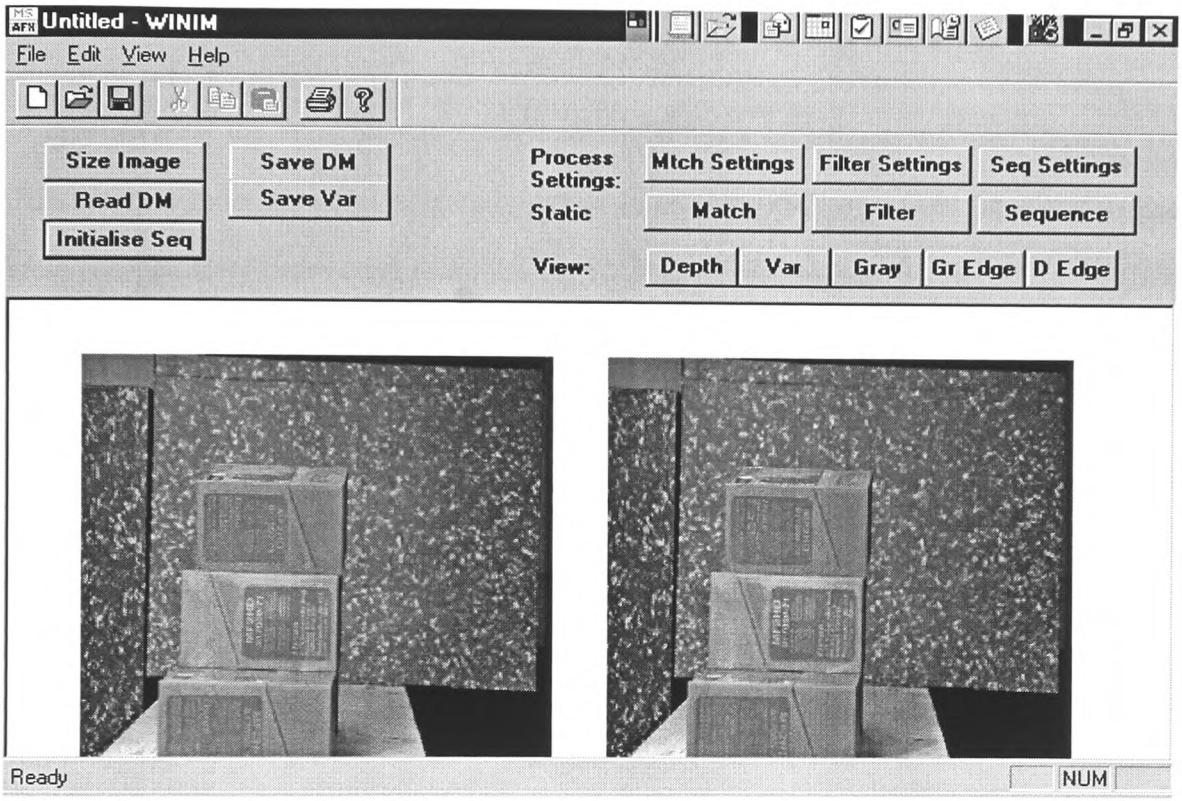

Figure 7.1: Screenshot of WINIM software main window

7.3.2 Simulated Signal Source

The following desirable characteristics were aimed for in the design of a source of simulated one-dimensional signals to train and test fuzzy filters:

(a) The source must be capable of generating many non-identical signals. This is to enable any filters which are created to be tested on different signals to ensure that the filters are not over trained so as to only be effective on one particular signal.

(b) The different signals generated by the signal source must share common properties. These properties should be such that the signal is a one-dimensional version of typical two-dimensional depth maps. In particular the depth gradients should be more likely to be low except at finite many discontinuities.

In order to meet the requirements a MATLAB script file program was written, 'CHAINGEN'. This program generates a signal made up out of an alphabet of five possible sub-signals. These are a constant level (horizontal part), a negative or positive discontinuity (vertical parts), and a positive or negative slope (sloping parts). The probability of the next part of the signal being one of these sub-signals depends on the sub-signal type of the current part of the signal. These probabilities are given by a state transition matrix whose entries tend to favour continuity in the signal, whilst allowing finite many discontinuities. The signal amplitude varies from zero to one.

A parameter can be passed to the signal generating function that defines a 'characteristic length' for the slope and horizontal sections of the signal. This characteristic length parameter is used to scale the signal. This is illustrated in figures 7.2 and 7.3, which show a length 100 signal, and a length 1000 signal. Using the default characteristic length the features of both signals appear at the same scale when plotted over their full length. Figure 7.4 shows a length 1000 signal with a characteristic length of 100 plotted over the full 1000 points, whilst figure 7.5 shows the section from sample numbers 300 to 399 of the same signal. The characteristic length parameter allows the generator to generate long runs of signal at a given constant scale. These long runs of signal can then be used as training and test data. The gradients of the slope sections of the signal are normally distributed about zero with a standard deviation set so as to discriminate against steep depth gradients. The gradients of the slope sections can also be set

to a constant value to produce a more restricted set of test data. An example of a constant gradient test signal is shown in figure 7.6.

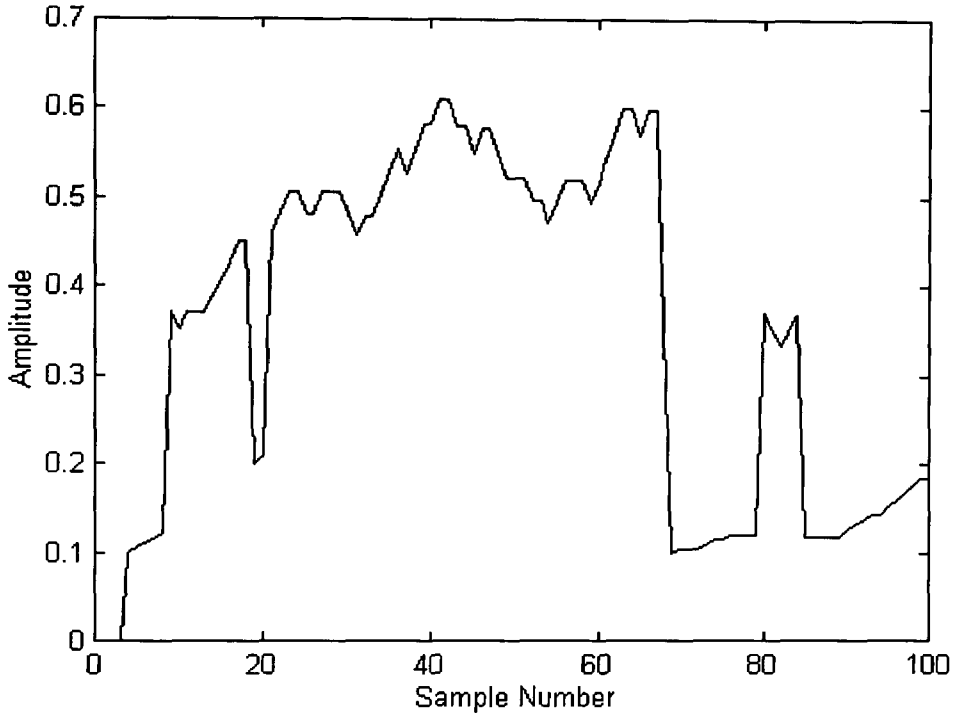

Figure 7.2: Signal of length 100 generated by 'CHAINGEN'

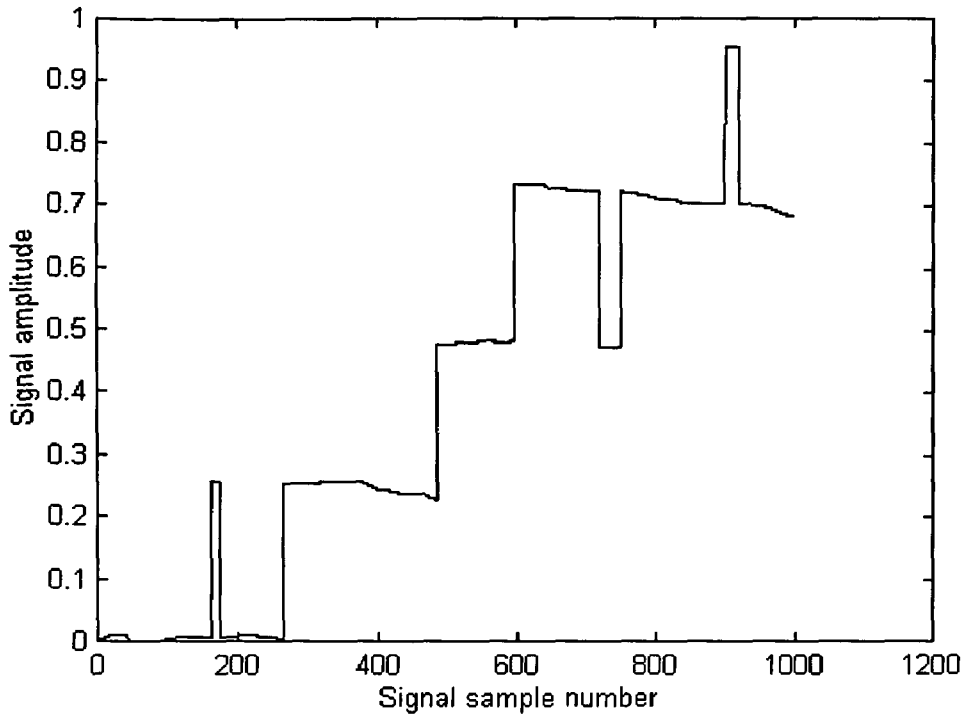

Figure 7.3: Signal of length 1000 generated by 'CHAINGEN'

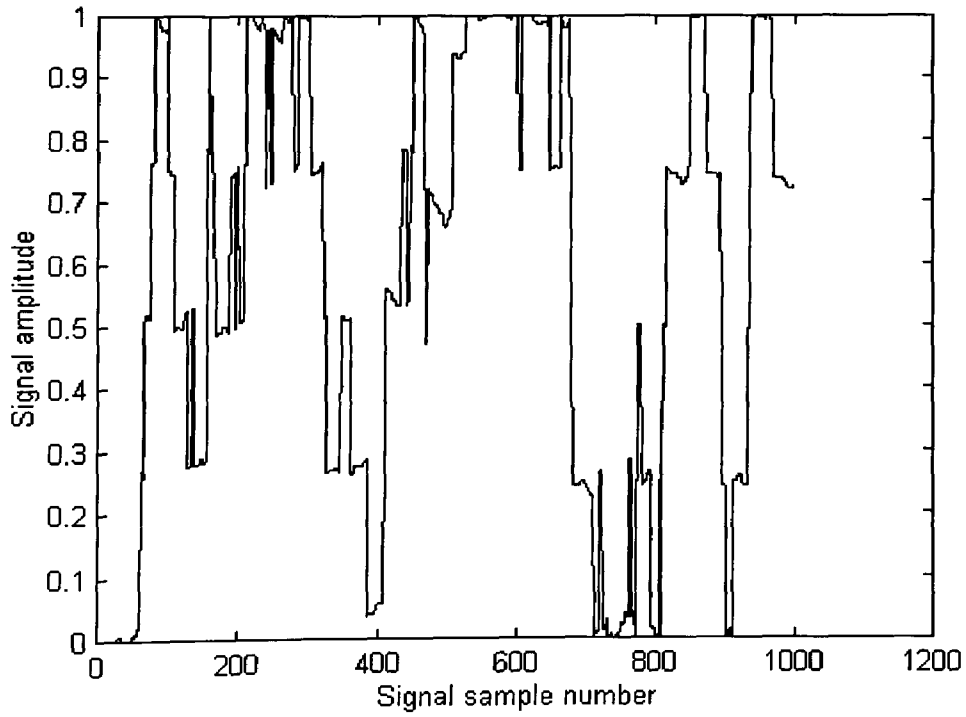

Figure 7.4: 1000 samples of Signal of characteristic length 100 generated by 'CHAINGEN'

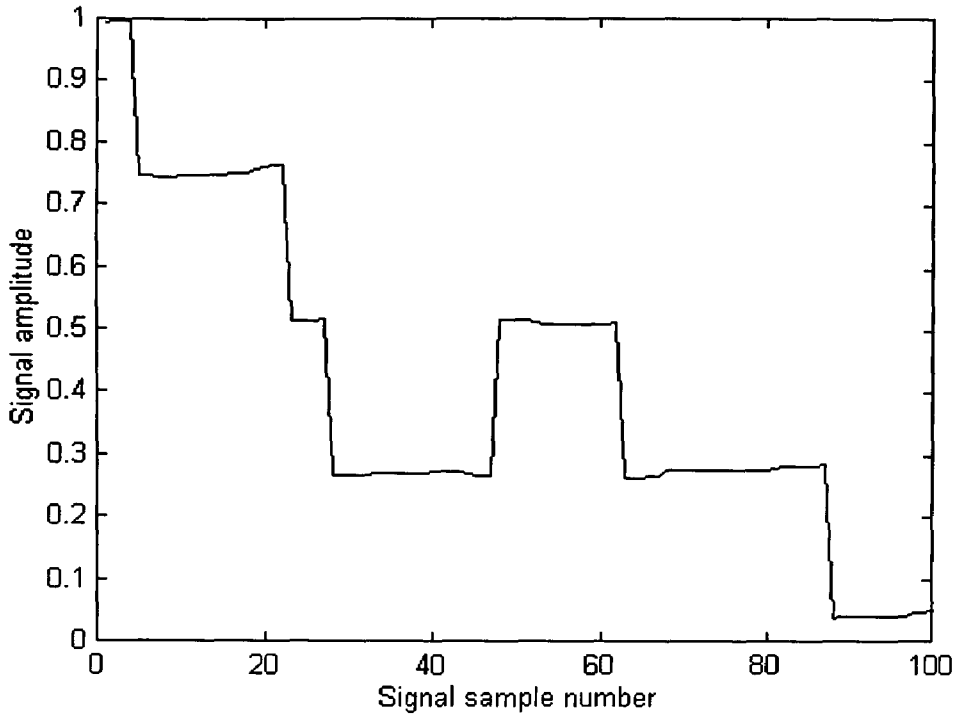

Figure 7.5: 100 samples of length 1000 signal of figure 7.4

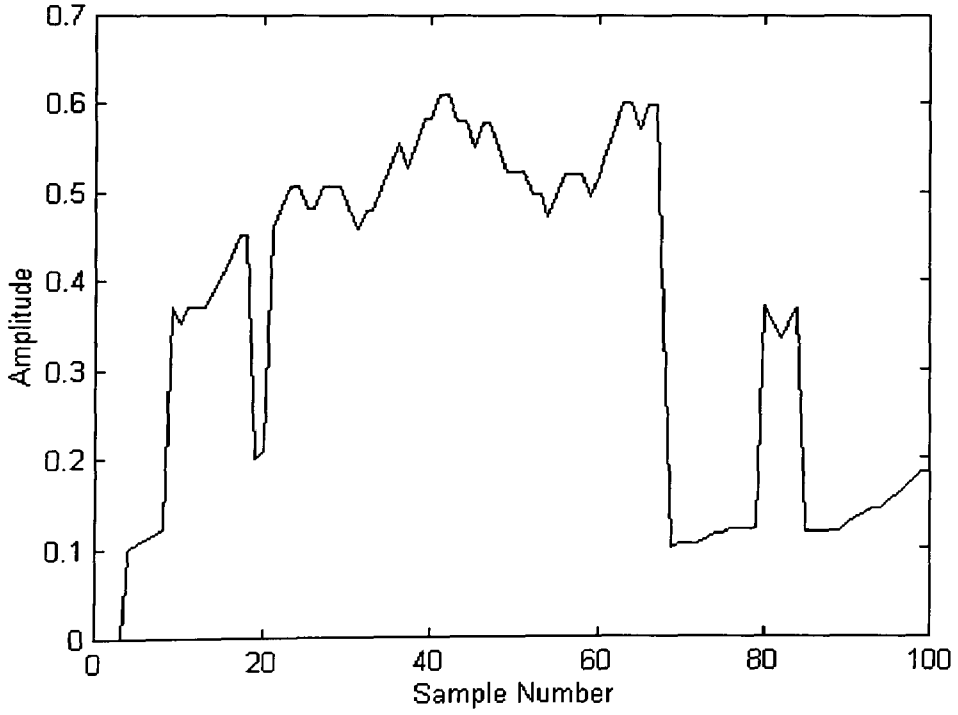

Figure 7.6: Example of test signal with constant gradient produced by 'CHAINGEN'

7.3.3 Source of noise to corrupt the signal

In order that the test signals should be one-dimensional versions of typical depth maps obtained from a correlation based matcher it was necessary to corrupt the signals generated by CHAINGEN with additive impulsive, Gaussian, or mixed impulsive/Gaussian noise. The addition of Gaussian noise was achieved using the MATLAB 'Randn' command, which generates vectors of Gaussian distributed numbers of unity standard deviation. In order to generate impulsive noise a MATLAB script file program called 'IMPTRAIN' was created. IMPTRAIN generates a vector of impulses and has the calling syntax:

function imptrn = IMPTRAIN(trainlen, occurrence, ampdist)

The parameter *trainlen* defines the length of the impulse train. *Occurrence* defines the percentage occurrence of the impulses. *Ampdist* can take the values '*flat*' or '*gaussian*' which give uniformly or Gaussian distributed amplitudes for the impulses. Figure 7.7 shows a typical length 100, 10% occurrence impulse train generated by IMPTRAIN.

Figure 7.8 shows a mixed Gaussian and impulsive noise signal. The histogram for this mixed Gaussian and impulsive noise is shown in figure 7.9. This can be seen to be very similar in form to the histograms of the disparity maps produced using correlation-based matching shown in figures 3.28 and 3.29.

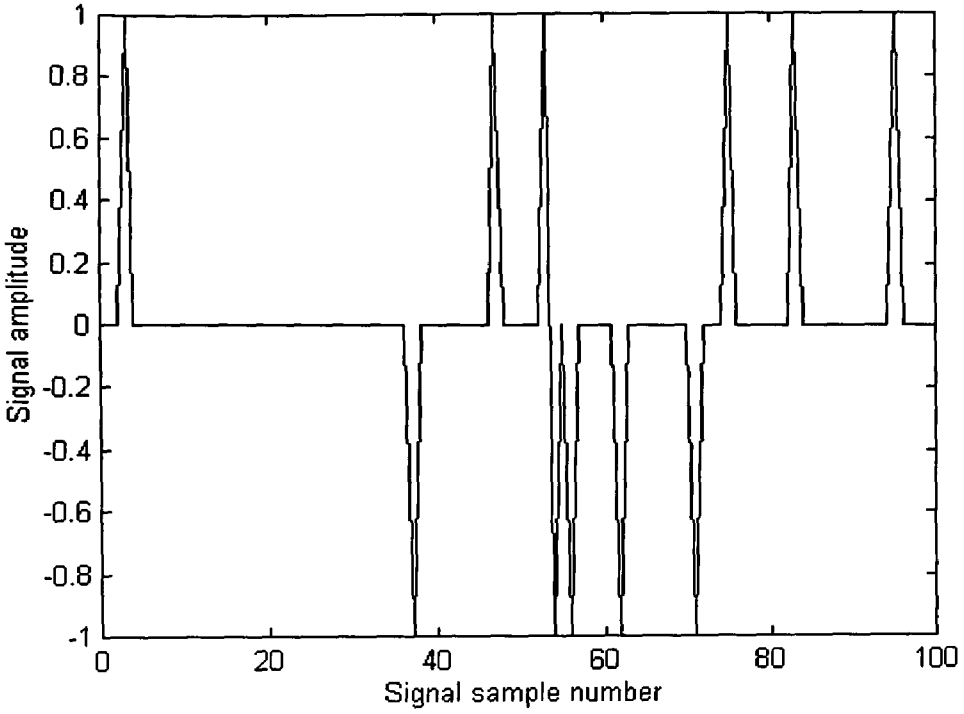

Figure 7.7: Typical length 100, 10% occurrence impulse train generated by IMPTRN.

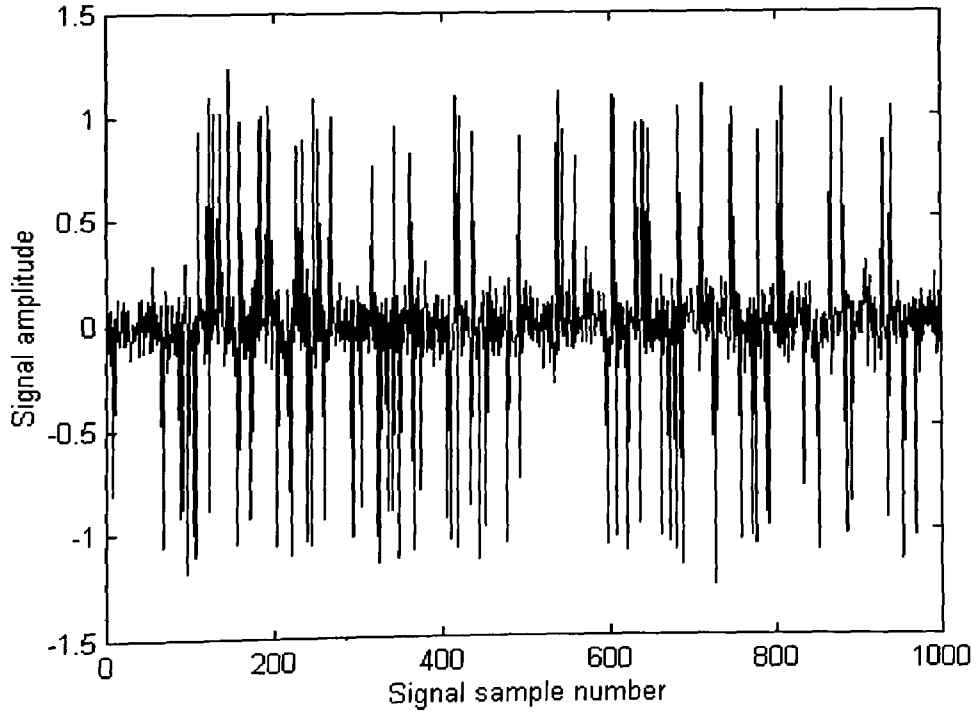

Figure 7.8 Mixed Gaussian (standard deviation 0.1) and impulsive (10% occurrence) noise

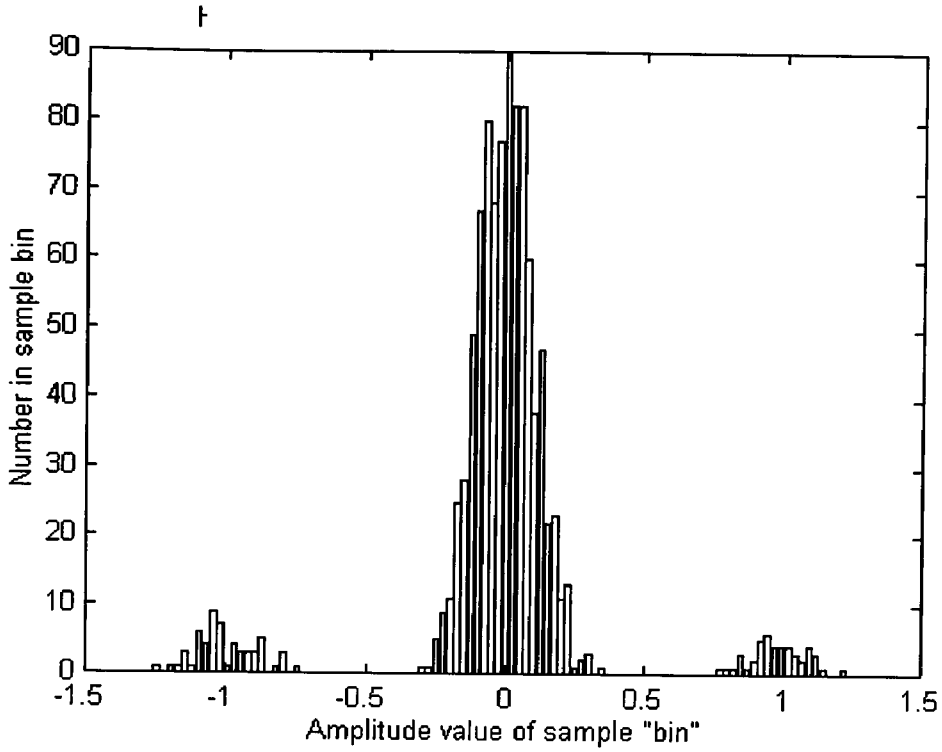

Figure 7.9: Histogram of noise signal of figure 7.8

7.4 Approximation of median filter using a Fuzzy Filter

7.4.1 Motivation for approximating a median filter

It is trivial to achieve a linear filter using the FIR type Sugeno filter architecture since the output sets of the Fuzzy system are linear filters acting on the data. However, as discussed in section 4.6.1, the median filter is a useful non-linear filter that preserves edges and is effective in removing impulsive noise. If a fuzzy filter is to share these characteristics then it should be able to approximate a median filter's behaviour. A set of rules for such an approximation can be determined heuristically from knowledge of the behaviour of the median filter. This use of prior knowledge demonstrates one of the advantages of fuzzy function approximation over neural network function approximation.

7.4.2 Heuristic and trained median approximators

A set of rules and corresponding output sets for a fuzzy approximation to a three element median filter with three input fuzzy sets per input is shown in table 7.1 (in the notation described in section 5.3.4). The rules are a subset of six out of the possible 27 rules for such a system. In addition a 'hidden ELSE' rule as described in section 5.11.1 is included in the rulebase. The hidden ELSE rule simply outputs the second value of the input vector if none of the rules in the rulebase is fired by the input vector. The initial input fuzzy set parameters can be chosen such that the Gaussian input set membership functions are evenly distributed across the input universe of discourse. These initial input fuzzy sets are shown in figure 7.10.

Rule number	Rule	Output set parameters
1	[1,2][2,1][3,3]:1	1,0,0
2	[1,2][2,3][3,1]:2	1,0,0
3	[1,1][2,2][3,3]:3	0,1,0
4	[1,3][2,2][3,1]:4	0,1,0
5	[1,1][2,3][3,2]:5	0,0,1
6	[1,3][2,1][3,2]:6	0,0,1
Hidden ELSE Rule	ELSE	0,1,0

Table 7.1: Rulebase and output sets for a fuzzy approximation to a three element median filter

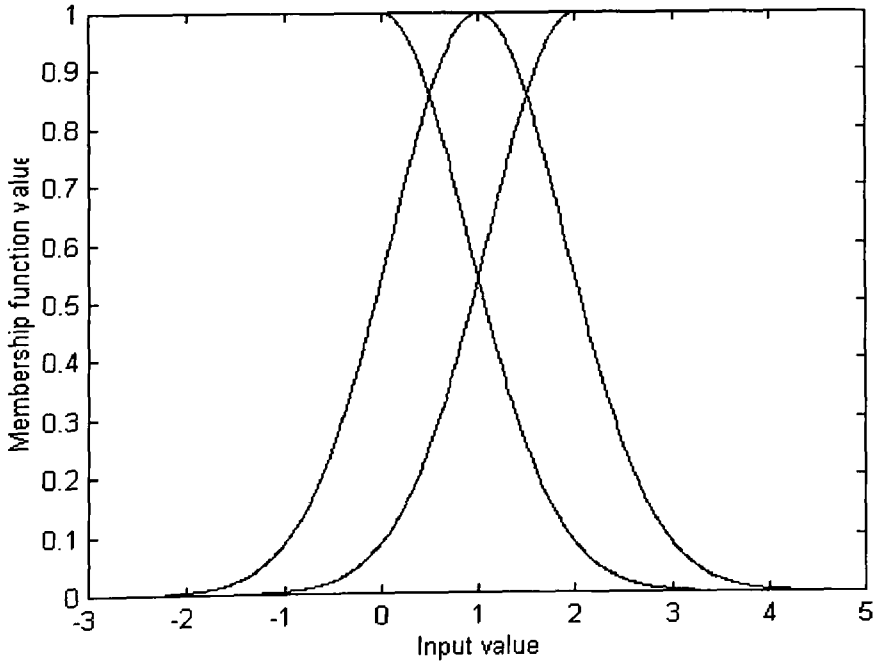

Figure 7.10: Initial input fuzzy set membership functions for fuzzy approximation to a three element fuzzy filter (the input sets are initially the same for all inputs)

In addition to the heuristic six rule approximation to the median, fuzzy systems with 4, 9, 10, 16 and 27 rules were generated using the 'initialise system' (section 5.11.2). Using MATLAB, a set of 200 training data pairs was created. These pairs consisted of inputs that were triples of evenly distributed random numbers in the range [0,3] and outputs that were the median of the input triples. These training data pairs were used to train the six fuzzy systems to approximate the median using the combined simplex, simulated annealing, and linear least squares algorithm described in section 5.10. In order to test the trained fuzzy approximators' ability to generalise from the training data, they were applied as a three-element filter to a test signal of 1000 samples. Their output was then compared with the output of a three element median filter. The test signal consisted of a set of 1000 evenly distributed random numbers. As a further comparison a naïve approximator, which simply repeated the centre value in the 3 element window, was applied to the test data.

7.4.3 Tests and results

The mean square error (MSE) over the training data set for the untrained six rule fuzzy filter was 0.035. After training, a MSE of 0.0092 was achieved over the training data set. The results for this and the other filters are summarised in table 7.2.

The results summarised in table 7.2 appear to show that the greater the number of rules the lower the MSE achieved on training data. The number of rules versus the MSE on training data is plotted in figure 7.11. The 27-rule (exhaustive) fuzzy system in particular achieved a better MSE as a median approximator acting on the training data compared with the six rule fuzzy approximator. However, it failed to maintain this improvement in performance on both the uniformly distributed test data and the test signal. In general, the performances of the

approximators with fewer rules are degraded less on the test data than the performances of those with more rules.

Approximator type	MSE on training data	MSE on test data
Naïve approximator	N/A	0.53
Untrained heuristic 6-rule	0.035	0.0345
Trained 1-rule	0.0838	0.1056
Trained 4-rule	0.026	0.0338
Trained 6-rule	0.00941	0.0173
Trained 9-rule	0.014	0.0209
Trained 10-rule	0.0105	0.0118
Trained 16-rule	0.00425	0.0119
Trained 27-rule	0.00265	0.0252

Table 7.2: MSE for training data and test data for 3-element median approximators

The trained and untrained approximators were also applied as a three element filter to signals generated using CHAINGEN and corrupted by noise and their output compared with that of a three element median filter. These two outputs for the median and trained filter are shown plotted on the same graph in figure 7.12. On this graph, it is difficult to see the difference between the approximator and the median filter. Therefore, the difference between the median and the approximator is plotted in figure 7.13. The corresponding graphs for the untrained fuzzy approximator are shown in figures 7.14 and 7.15. The same test was applied to the 27-rule approximator and its performance was qualitatively similar to those shown in figures 7.12 to 7.15.

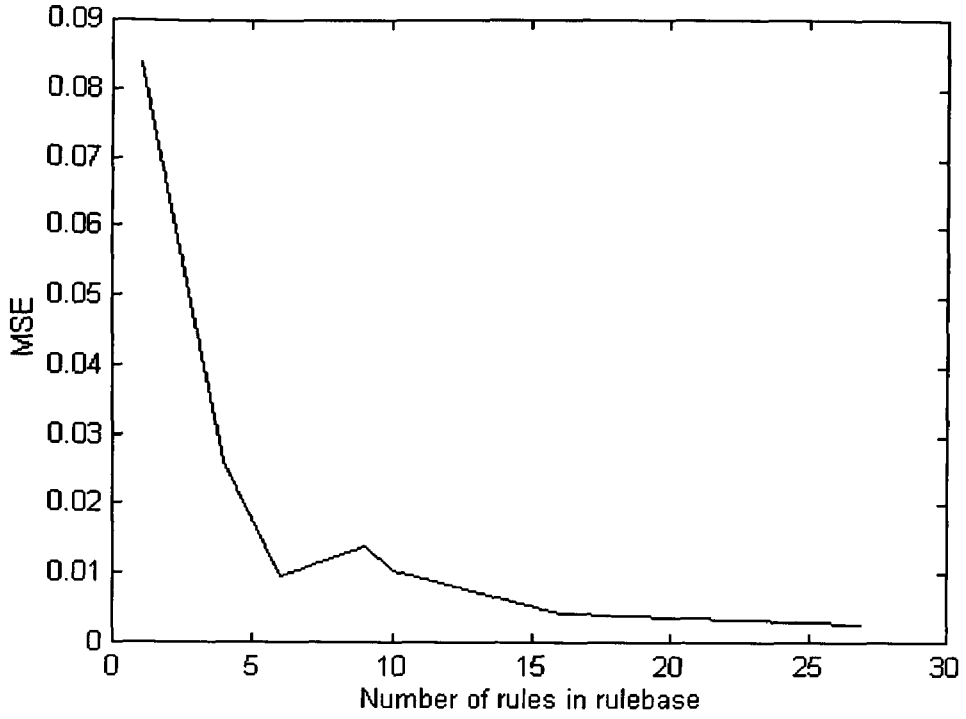

Figure 7.11: Plot of MSE versus number of rules for fuzzy approximator to median.

A 20-rule approximation to a five element median filter was also generated and tested. This approximator achieved a MSE of 0.008 on training data and 0.115 on the test data. The MSE for the fuzzy approximators on the signal generated by CHAINGEN are shown in table 7.3

Type of approximator	Mean squared error on test signal generated by CHAINGEN
untrained 6 rule	0.034
trained 6 rule	0.031
trained 27 rule filter	0.053
trained 5 element, 20-rule filter	0.49

Table 7.3: MSE for the fuzzy approximators on the signal generated by CHAINGEN

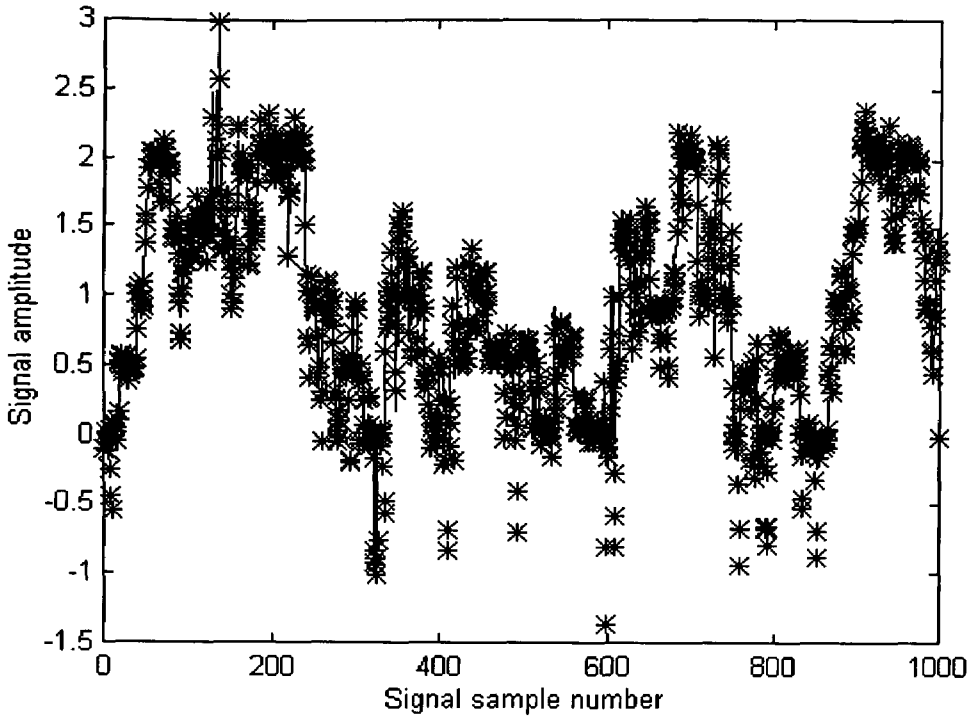

Figure 7.12: Plots of test signal after filtering with median (-) and trained fuzzy approximation to median (*) filters.

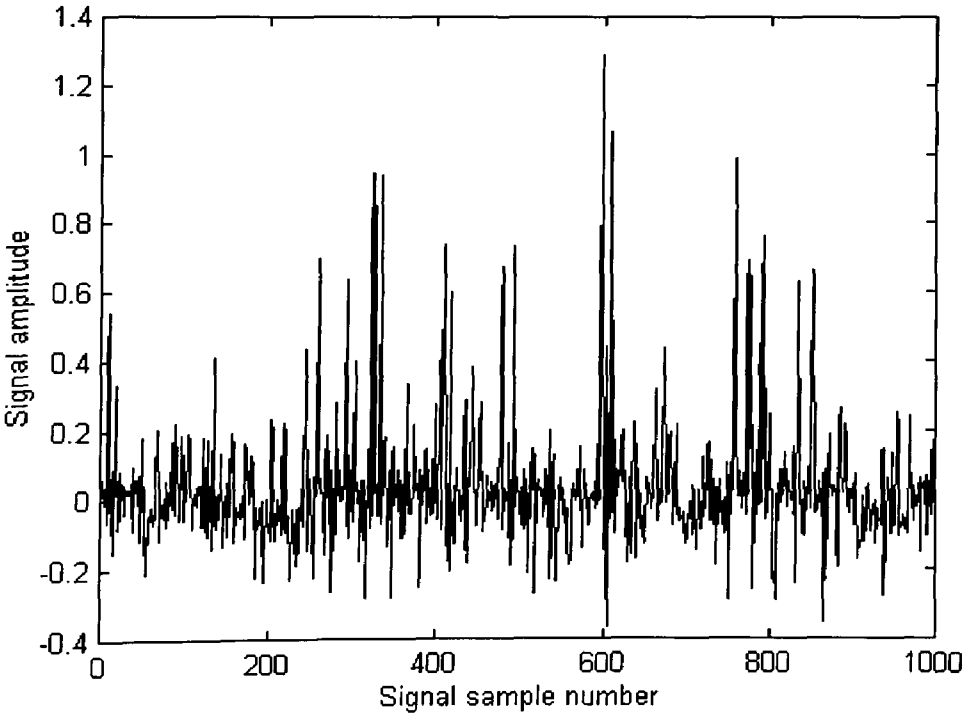

Figure 7.13: Plot of difference between test signal after filtering with median and trained fuzzy approximation to median filters.

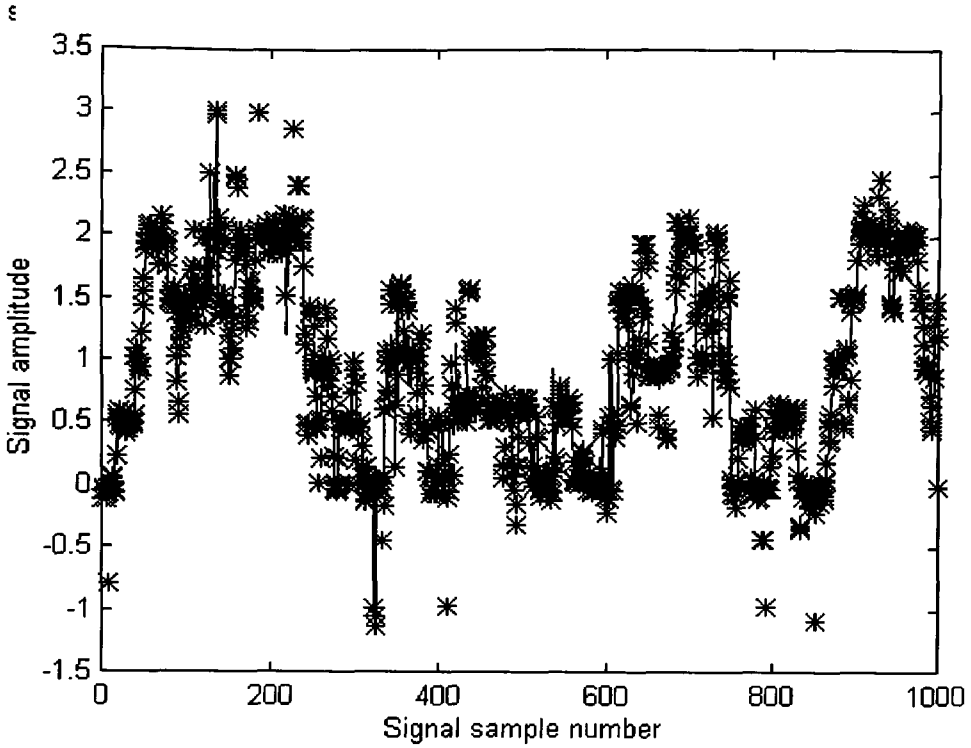

Figure 7.14: Plots of test signal after filtering with median (-) and untrained fuzzy approximation to median (*) filters.

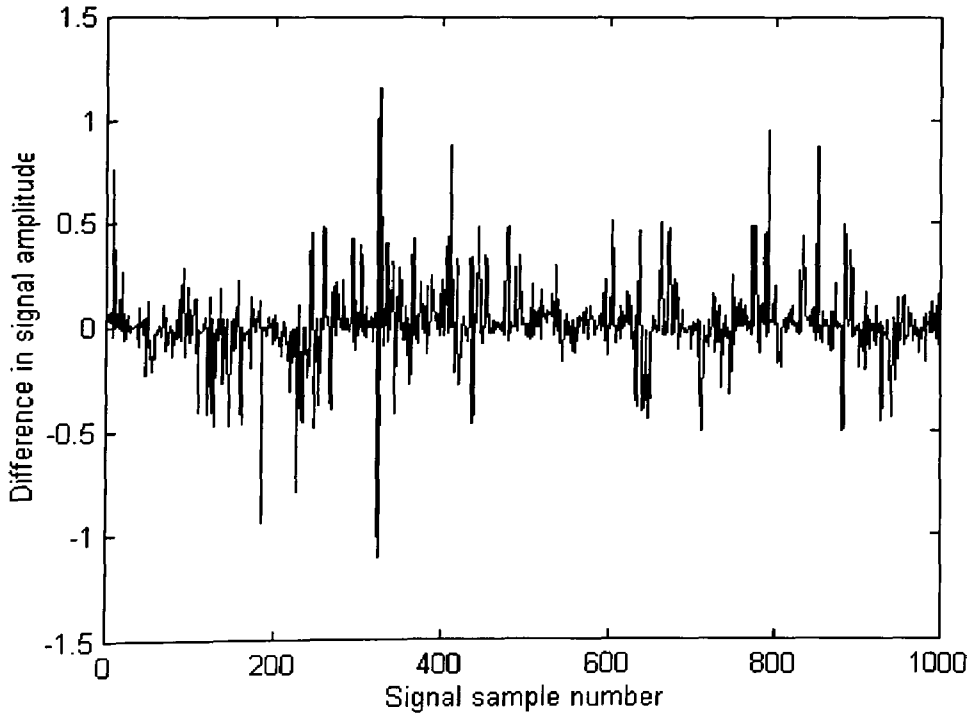

Figure 7.15: Plot of difference between test signal after filtering with median and untrained fuzzy approximation to median filters

7.4.4 Discussion

It is interesting to note that fuzzy systems are often tested on chaotic time series prediction (Jang, 1993). For such systems the benchmark with which the outputs of the fuzzy system are compared is the output of a first order hold. In testing the effectiveness of a fuzzy approximator to the median it is difficult to identify a reasonable 'naïve' approximator with which to compare the performance of the fuzzy approximator. This may explain why fuzzy prediction is often used in the literature as an example of fuzzy function approximation. For a fuzzy median approximator operating as a median filter one possibility is to take the unfiltered raw signal as the most naïve approximator to the median filter and using its output as a benchmark. However, whether this or the raw mean squared error is used as a comparison, the performance of fuzzy approximators when used on test data after training depends on the similarity between the training data and the verification or test data.

The dependence of the trained system's performance on the training data is both an advantage and disadvantage in the context of fuzzy filters. It is an advantage if filtering is viewed as a process in which *a priori* assumptions about the underlying data and noise are applied to the noisy signal in order to remove the noise. The fact that trained filters closely model the training data allows such filters to capture the prior model from the training data. The disadvantage lies in the problem of ensuring that the training data is a true representation of the real signals and noise for the problem which is being modelled. It could be argued that the results of table 7.2 suggest that the fuzzy systems with more rules and which appear to model the training data best were in some way over-trained on that data and so were unable to effectively generalise to the test data. The interesting question of over-training, in which a trained system is said to model the training data too closely to generalise well to other data, is related to the question of identifying sufficiently large and representative training data sets. The one-dimensional training data generated by CHAINGEN and IMPTRAIN is designed to address the problem of

representative data for depth and disparity maps. The question of sufficiency in the training data not only depends on the underlying signals and noise being modelled, but also depends on the scaling of the fuzzy system (see section 5.11.1). Such sufficiency questions represent interesting opportunities for further work.

7.5 Three-element Fuzzy Filters

7.5.1 Introduction

The work of section 7.3 was concerned with approximating a median filter. In fact, what is desired is a filter that exceeds the performance of a median, moving average or practical Wiener filters. As noted in section 7.3.4, the performance of a fuzzy approximator depends on the adequacy of the training data. In order to test the concept of using fuzzy filters whilst avoiding the question of training data adequacy some constraints are applied in this section to the test and training data. The CHAINGEN program described in section 7.3.2 allows the generation of strongly similar but non-identical signals. Thus training data generated using CHAINGEN and IMPTRAIN will be strongly representative of any test data generated by these programs. This section describes work to train and test three-element fuzzy filters which outperform both median and moving average filters on this test data and match or outperform ideal Wiener filters. However, the performance of these ideal Wiener filters is not representative of what would be obtained by a practical Wiener filter. This is because they use exact knowledge of the signal and noise power spectral densities, which would have to be estimated by a practical Wiener filter, but they represent a firm benchmark with which to compare the fuzzy filter.

7.5.2 Construction of filters

The nine-rule filter contains the same six rules as outlined in table 7.1 with the addition of the rules and output sets shown in table 7.4. This nine-rule filter was then trained on a set of 998 samples of training data that was generated by CHAINGEN using the constant gradient option, with a characteristic length of 100, and corrupted by IMPTRAIN. The impulse-corrupted training data was arranged in triples as input training data and the uncorrupted CHAINGEN output corresponding to the centre sample of the corrupted data triples was taken as the output training data. Before training, the nine-rule filter had a MSE of 0.0747 over the training data set. It achieved a MSE of 0.0286 after training.

Rule number	Rule	Output set parameters
1 to 6	As table 7.1	As table 7.1
7	[1,1][2,1][3,1]:7	0.33,0.33,0.33
8	[1,2][2,2][3,2]:8	0.33,0.33,0.33
9	[1,3][2,3][3,3]:9	0.33,0.33,0.33
Hidden ELSE Rule	ELSE	0,1,0

Table 7.4: Rulebase and output sets for nine rule fuzzy filter

In addition, using the same training data, several nine-rule filters, a 16-rule filter and a 27-rule filter were generated using the 'initialise system' option of FTEST and combinations of the training techniques described in sections 5.10 and 5.11.

7.5.3 Tests on three-element filters

All the three-element filters were tested in WINIM on the same 99-sample test signal. This test signal was generated using CHAINGEN with the constant gradient option and corrupted using IMPTRAIN. The uncorrupted version of this signal is shown in figure 7.16, whilst the signal corrupted by a 20% occurrence impulse train is shown in figure 7.17. For comparison purposes, three-element median and moving average filters were also applied to the test data as

well as a Wiener filter. The Wiener filter was separately implemented in the frequency domain using MATLAB.

The results from the training and test data for these filters is summarised in table 7.5. Figures 7.18 to 7.20 Show the results of applying the median, moving average, and Wiener filters to the test data. Figure 7.21 and figure 7.22 show the effects of applying the fuzzy filters 1d3rg5imp.fis and 27rulet3.fis to the test data.

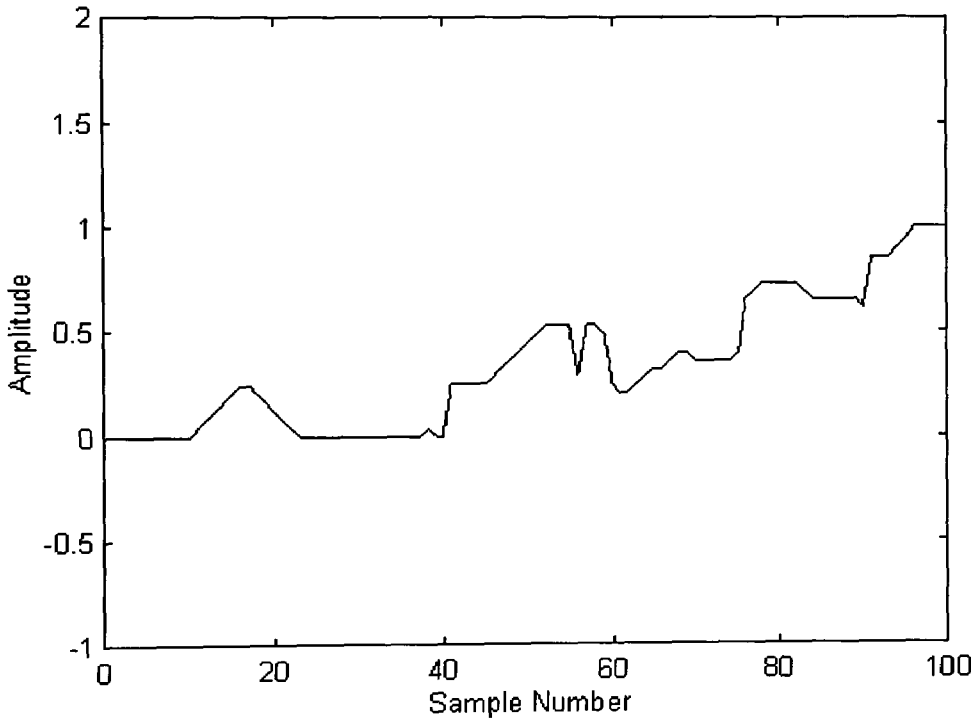

Figure 7.16: Uncorrupted test data used to test three-element filters.

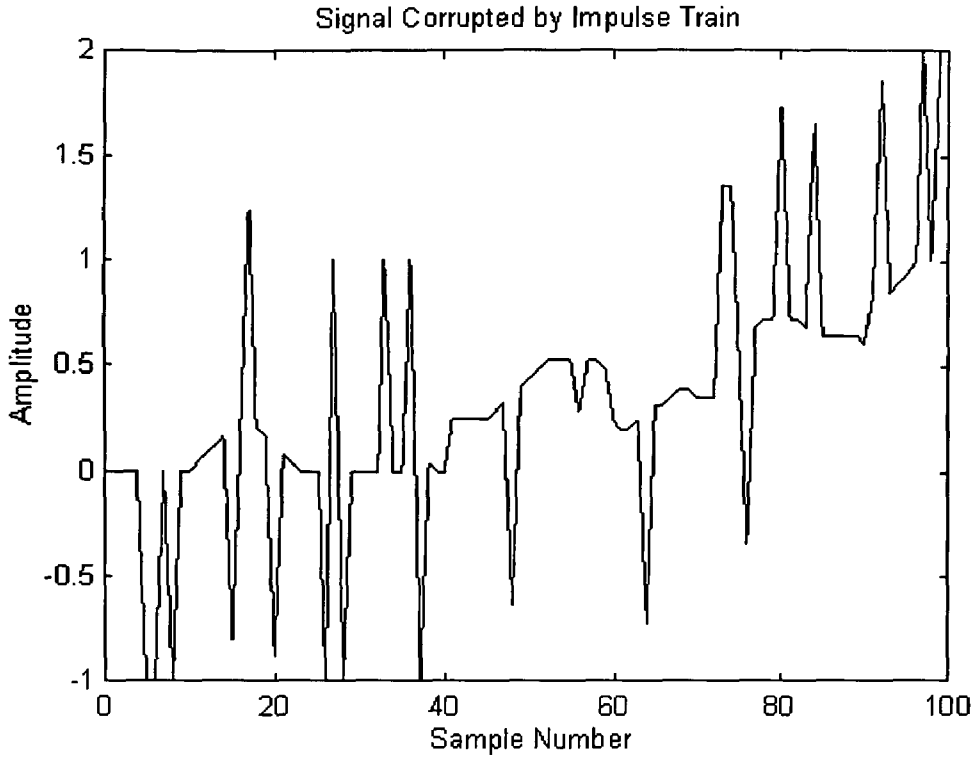

Figure 7.17: Test data of figure 7.16 corrupted by 20% occurrence impulse train

Name of filter	Description	MSE on training data	MSE on test data
Wiener	Ideal Wiener filter in frequency domain	N/A	0.0158
Median	3 element median	N/A	0.0727
Moving average	3 element moving average	N/A	0.0734
3eleref.fis	Untrained 9 rule fuzzy system	0.0376	0.0747
3elerefrt	Trained version of 3eleref	0.01415	0.0286
1d3rg1imp.fis	9 Rule generated and trained fuzzy system	0.0210	0.0859
1d3rg3imp.fis	As 1d3rg1	0.0093	0.0157
1d3rg5imp.fis	As 1d3rg1	0.0082	0.0115
16rulegt.fis	16 Rule generated and trained fuzzy system	0.00965	0.0251
27rulet1.fis	27 Rule generated and trained fuzzy system	0.0053	0.0320
27rulet3.fis	As 27rulet1	0.0055	0.0101

Table 7.5 MSE achieved on training and test data for three-element filters.

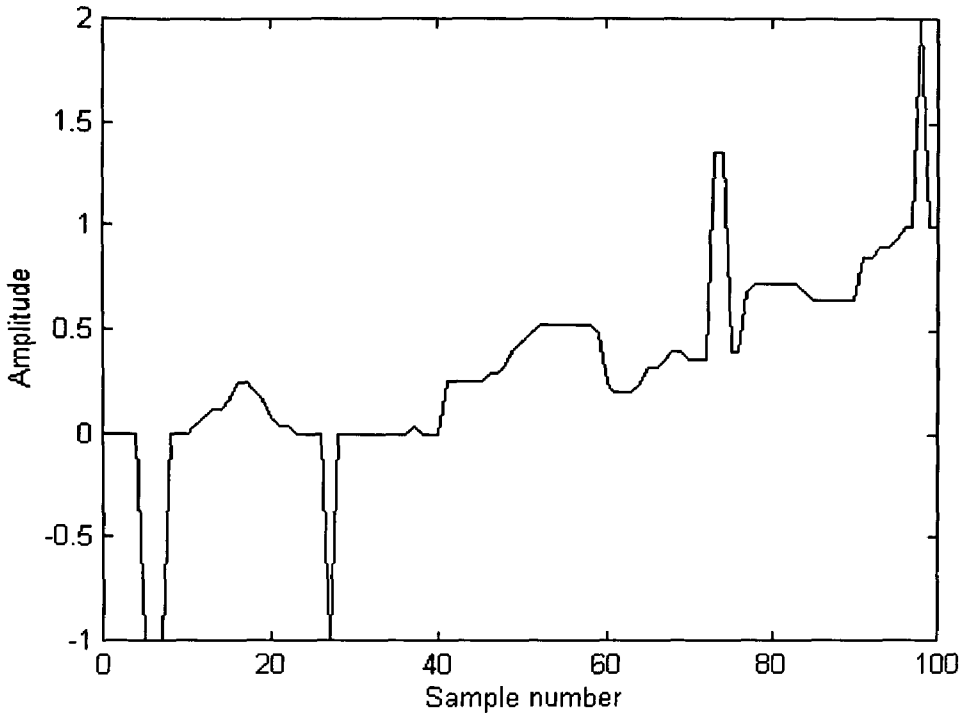

Figure 7.18: Test data after filtering with three-element median filter.

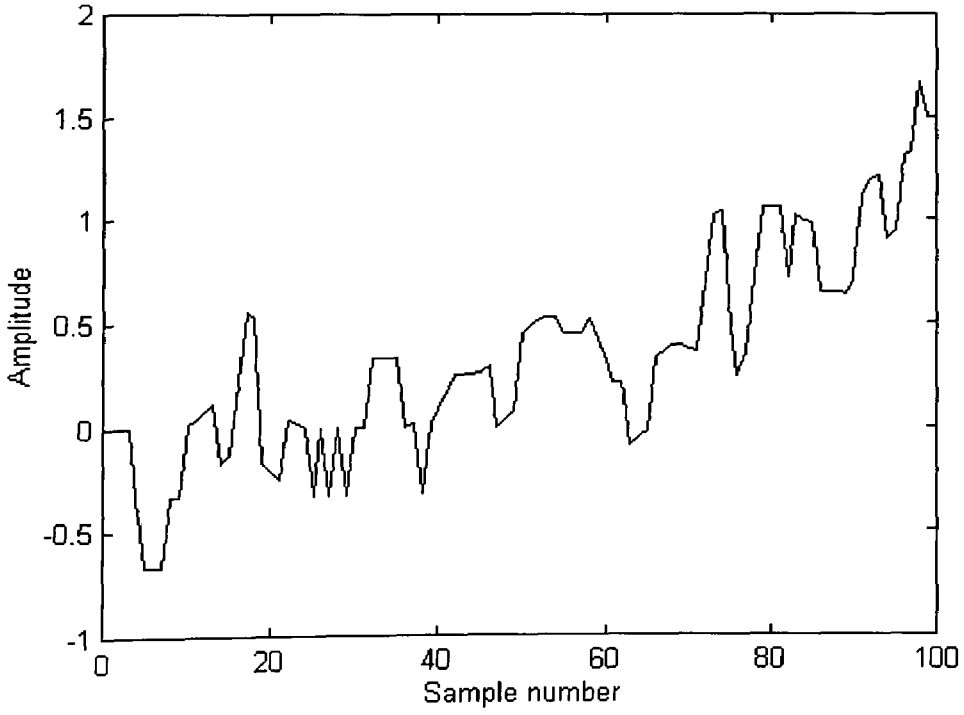

Figure 7.19: Test data after filtering with three-element moving average filter

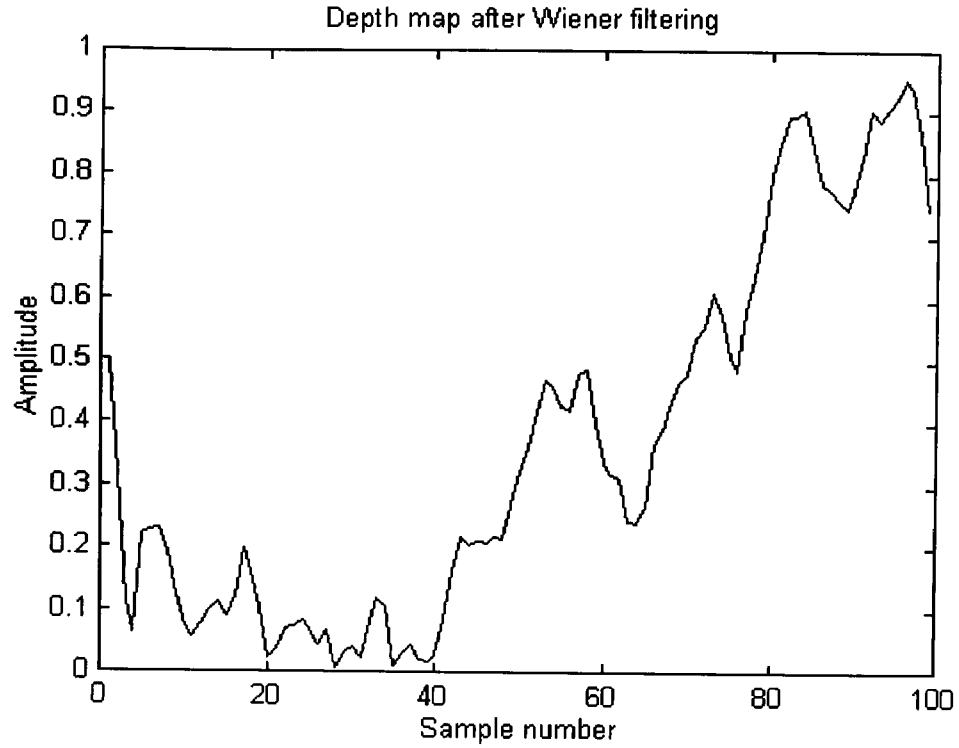

Figure 7.20: Test data after filtering with ideal Wiener filter.

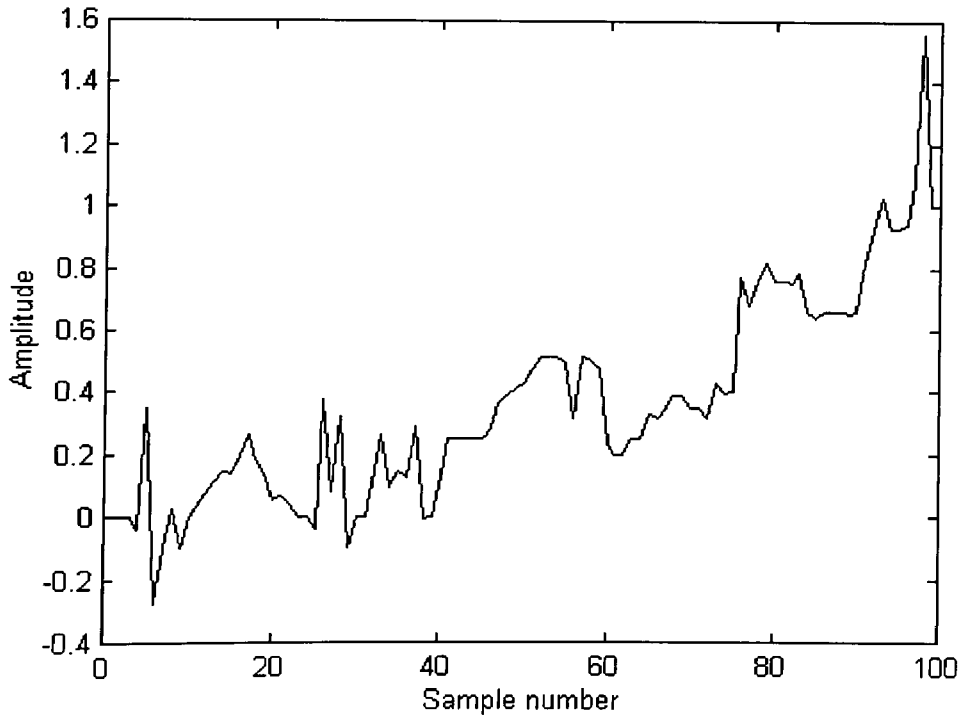

Figure 7.21: Test data after filtering with fuzzy filter 1d3rg5imp.fis

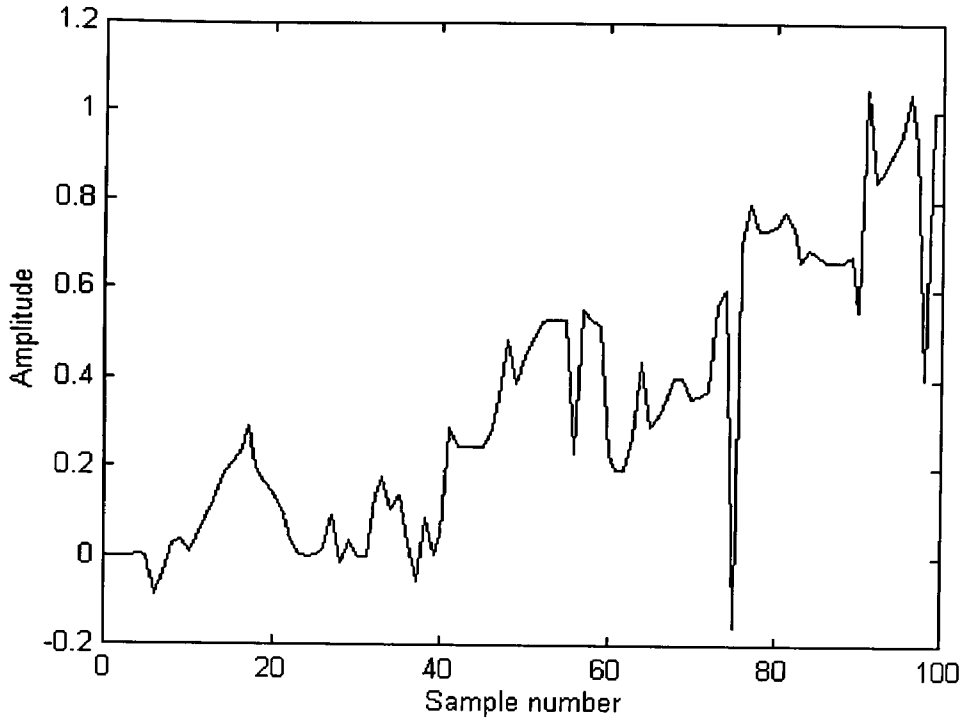

Figure 7.22: Test data after filtering with fuzzy filter 27rulet3.fis

A further test that was carried out was to generate a nine-rule three-input filter using the 'initialise system' option and to train this system on training data that was corrupted with mixed Gaussian and impulsive noise. The signal was generated by CHAINGEN, the impulsive noise component had a 20% occurrence and the zero-mean Gaussian noise had a standard deviation of 0.2. The inherent MSE of the noisy signal compared with the original clean signal was 0.2241. Again, the performance of the fuzzy filter was compared with those of median, moving average and an ideal Wiener filter. The Fuzzy system achieved a MSE of 0.0273 on the training data. The comparison between the fuzzy filter and the other filter types is given in table 7.6.

Type of filter	MSE on test signal
Ideal Wiener	0.0272
Moving average	0.0758
Median	0.1031
Nine-rule trained fuzzy filter	0.0465

Table 7.6: Comparison of filter MSE for mixed noise test signal.

7.5.4 Discussion

Three-element fuzzy filters have been trained and tested on signals corrupted by impulsive and mixed Gaussian and impulsive noise. The MSE performance of these filters was compared to that of median, ideal Wiener, and moving average three-element filters. The training and test signals, although different, were limited to a restricted class of signals generated by CHAINGEN. Within these restrictions the fuzzy filters, three of the trained fuzzy filters exceeded or equalled the MSE performance of the benchmark Wiener filter. Six of the trained fuzzy filters produced a lower MSE than the median or moving average filters on the test data. An untrained heuristically generated nine-rule fuzzy system produced a MSE that was close to that of the three-element median and moving average filter

7.6 Two-dimensional depth maps

7.6.1 Depth maps used to test fuzzy filters

Three types of two-dimensional signal were used to test the action of fuzzy filters on depth maps. The three types were a simulated depth map called 'ref', depth maps generated using stereo matching on a random dot image sequence 'cake', and depth maps produced from sequences of real images 'box' taken by a CCD camera. The advantage of the first two types of depth maps is that for these depth maps the 'ground truth' depth map is known and therefore

the performance of the fuzzy filters can be compared with that of a moving average and a median filter. The simulated depth maps are, however, less complex and do not model the distortions present in real images acquired by a camera.

The simulated depth map, 'ref', is simply a two-dimensional signal which can be corrupted by Gaussian or impulsive noise. It is shown uncorrupted in figure 7.23 and like all the depth maps shown in this chapter, it is plotted relative to a reference plane which is 3 m in front of the camera. The ideal depth map that should be generated from the random dot image sequence 'cake' is identical to the simulated depth map of figure 7.23. Each random dot image in the simulated image sequence consists of areas that are shifted by an amount that correspond to the disparity that would result from the three dimensional scene represented by figure 7.23 and a fixed set of camera parameters. The camera parameters that were used are the same as those used for the real image sequence and were a focal length of 25 mm, a pixel size of 4×10^{-5} m, and a shift along the camera array axis of 4 mm between images. The first two images in the sequence 'cake' are shown in figure 7.24 as they appear in the WINIM software. Figure 7.25 shows the depth map that results from applying the stereo matching algorithm described in section 2.3.3 and the SSD matcher discussed in Chapter 3 to the image sequence 'cake' as implemented in the WINIM software. In producing the depth map of figure 7.25, a fixed matching patch size of 5 x 5 was used, the Kalman filter was disabled and the disparity was tracked through a sequence of five images. The MSE that was achieved was 0.7224.

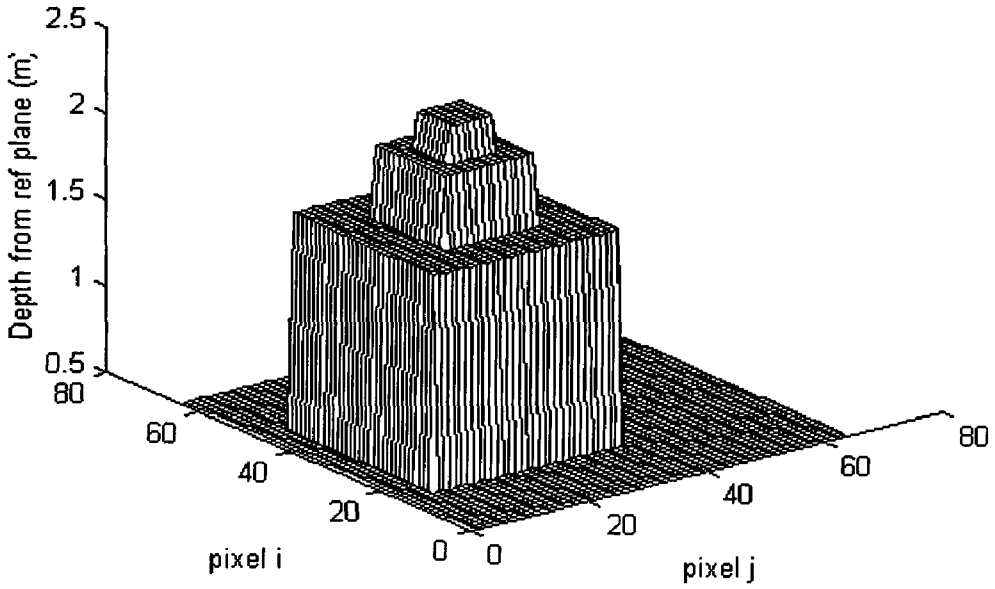

Figure 7.23: Simulated depth map 'ref'

With the Kalman filter enabled, and a WINIM option that varies the size of the SSD matching window also enabled, the depth map improved to that of figure 7.26. The variable SSD matching window option starts with a small matching window of 3 x 3 at each pixel and increases the window size up to a maximum window size if the uncertainty measure described in section 3.5 is above a threshold value. If the uncertainty measure is below this value the disparity is accepted and the matcher moves on to the next pixel. The MSE for figure 7.26 is 0.4653. The MSE for the depth map produced with a fixed matching patch size and the Kalman filter was 0.6582.

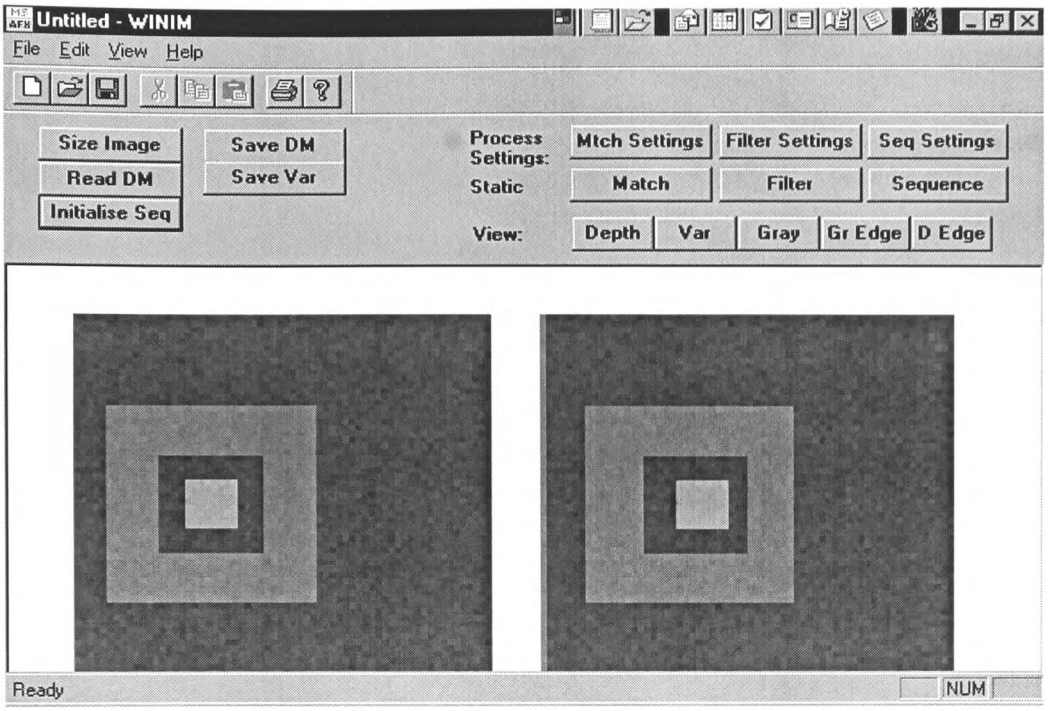

Figure 7.24: First two images of image sequence 'cake' as they appear in the WINIM software

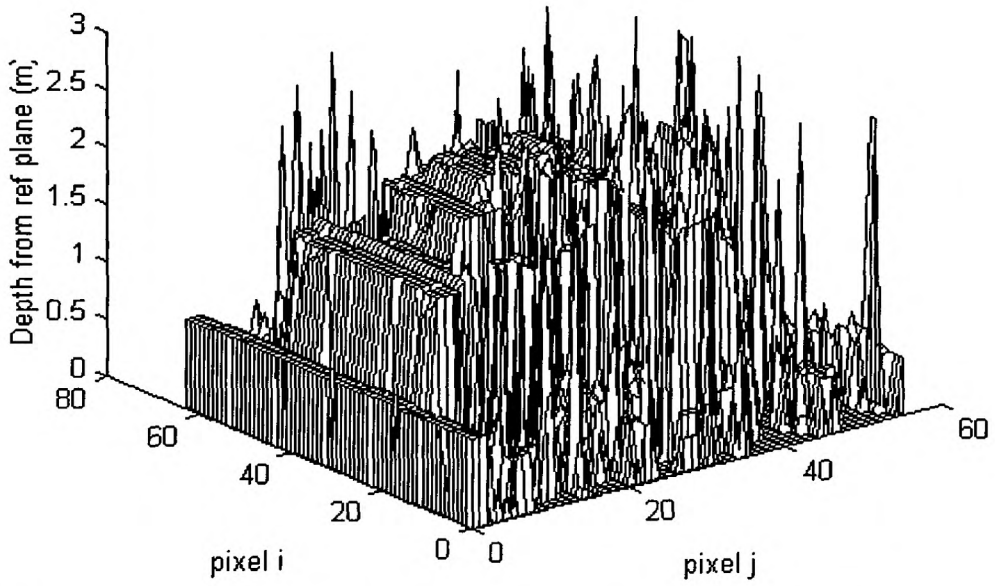

Figure 7.25: Depth map produced from noiseless simulated image sequence 'cake' with no filtering

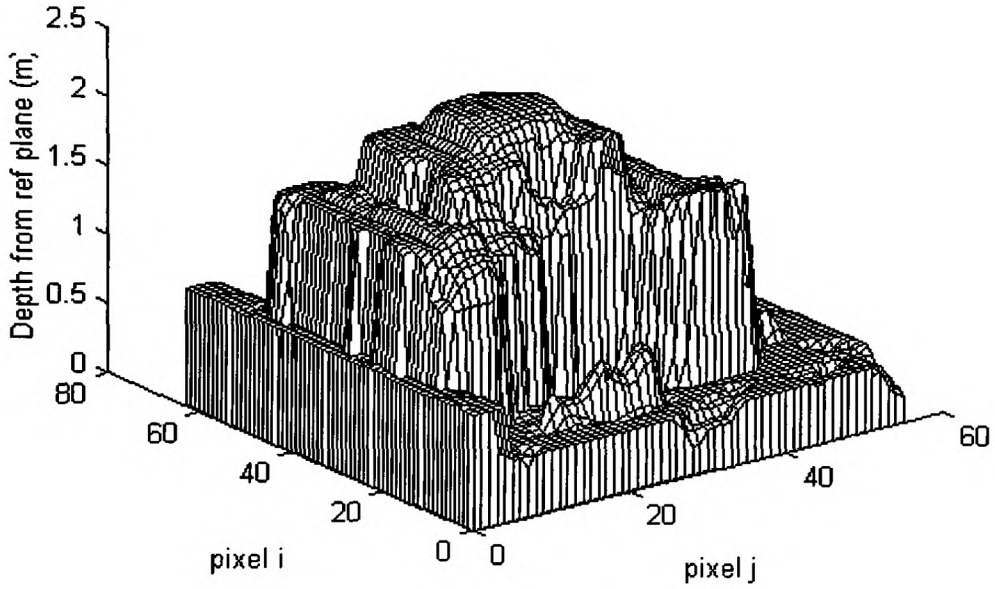

Figure 7.26 Depth map produced from noiseless simulated image sequence with Kalman filtering and variable SSD matching window size.

The first image of a sequence of real images 'box' is shown in figure 7.27. The images are of three computer disk boxes stacked on a wooden board and viewed against a speckled background. The depth map which results from these images with the variable patch size and Kalman filtering options enabled is shown in figure 7.28.

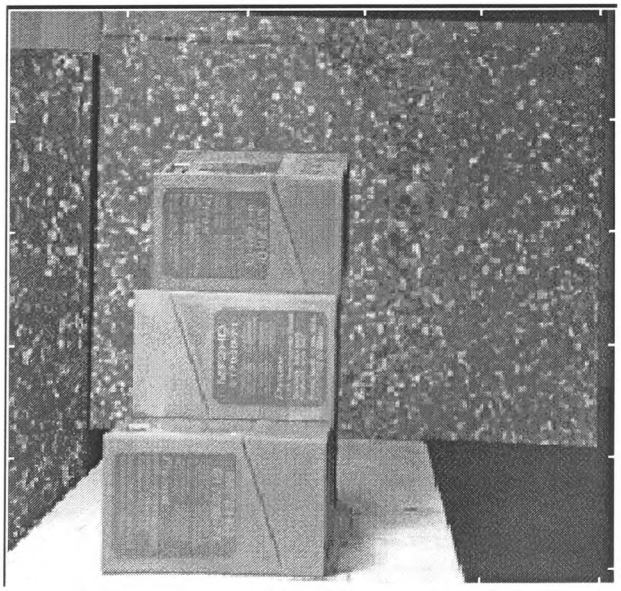

Figure 7.27: The first image of a sequence of real images 'box'

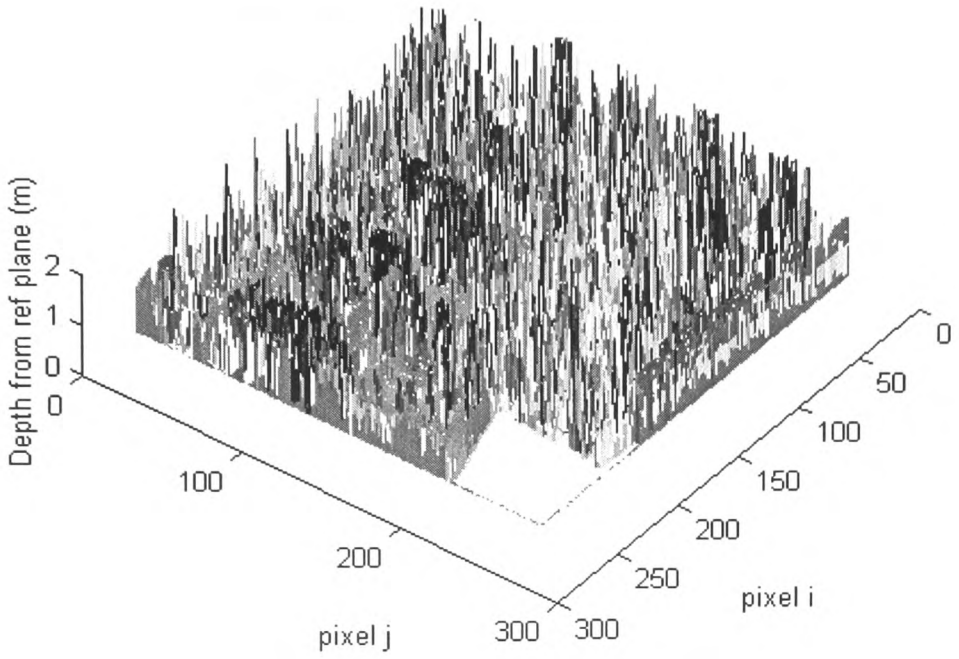

Figure 7.28: The depth map which results from image sequence 'box' with the variable patch size and Kalman filtering options enabled, but with no filtering applied between matching steps.

Figure 7.28 shows the extent of the potential filtering problem for depth maps generated from real image sequences using correlation based matching.

7.6.2 Extension of fuzzy filtering to two-dimensional depth maps

Because of the problem of rulebase explosion with dimensionality and the related problem of training high dimensional fuzzy systems discussed in section 5.11.1, the extension of the FIR type Sugeno fuzzy filter to two-dimensional windows is a severe computational challenge. For example the smallest two-dimensional window has nine elements, which with three fuzzy sets defined per input leads to an exhaustive rulebase of 3^9 =19683 rules. With two parameters per input set the nine element filter leads to a search space dimensionality of 18 for the input parameters. The time taken to train such a system with the current FTEST software is prohibitively long. Moreover, thus far all attempts at devising non-exhaustive rulebase filters both using a heuristic approach or using the rulebase training approach of section 5.11 have

failed to produce successful fuzzy filters. Therefore, the strategy that has been adopted to investigate the performance of fuzzy filters for depth map smoothing is to apply one-dimensional filters along the rows and then the columns of the depth map.

7.6.3 Filtering of depth maps using fuzzy filters

The depth map 'ref' was corrupted with impulsive noise generated by the MATLAB program IMPTRAIN. This noisy depth map was used as a test signal and filtered using a 3 x 3 moving average filter, a 3 x 3 median filter, and two fuzzy filters. The results are summarised in table 7.7.

Filter	MSE
Median	0.003
Moving Average	0.0217
9 rule fuzzy filter trained on mixed noise and signal generated by CHAINGEN	0.0333
27 rule fuzzy filter	0.0135

Table 7.7: Results of applying median, moving average, and two fuzzy filters to simulated depth map corrupted by impulsive noise.

Gaussian noise of different standard deviation was then added to the depth map and the resulting MSE was noted for the median, and moving average filters as well as the better of the two fuzzy filters. The results are shown as a plot of root MSE (RMSE) versus Gaussian noise standard deviation in figure 7.29.

It can be seen from figure 7.29 that the median is the better filter when the impulsive noise dominates, and that as the Gaussian component increases the moving average filter becomes the better filter. In no case is the fuzzy filter better than the median however, although the compromise made by the fuzzy filter between a median and a moving average filter can be seen. The target for an ideal fuzzy filter is for the curve of RMSE versus added Gaussian noise for the fuzzy filter to cross below that of the median.

Thus far, despite many attempts, it has proved impossible to train a fuzzy filter using fuzzy filters and the approach described in section 7.62 that exceeds the performance of the median filter on the simulated depth map 'ref' corrupted with mixed noise. This is illustrated in Figure 7.30, which shows a graph of RMSE versus the standard deviation of the added Gaussian noise for several fuzzy filters. The lower dashed line shows the performance of the median filter, which can be seen to form a lower bound. The upper dashed line is the RMSE before filtering, and shows the point at which a filter starts to worsen rather than improve the signal.

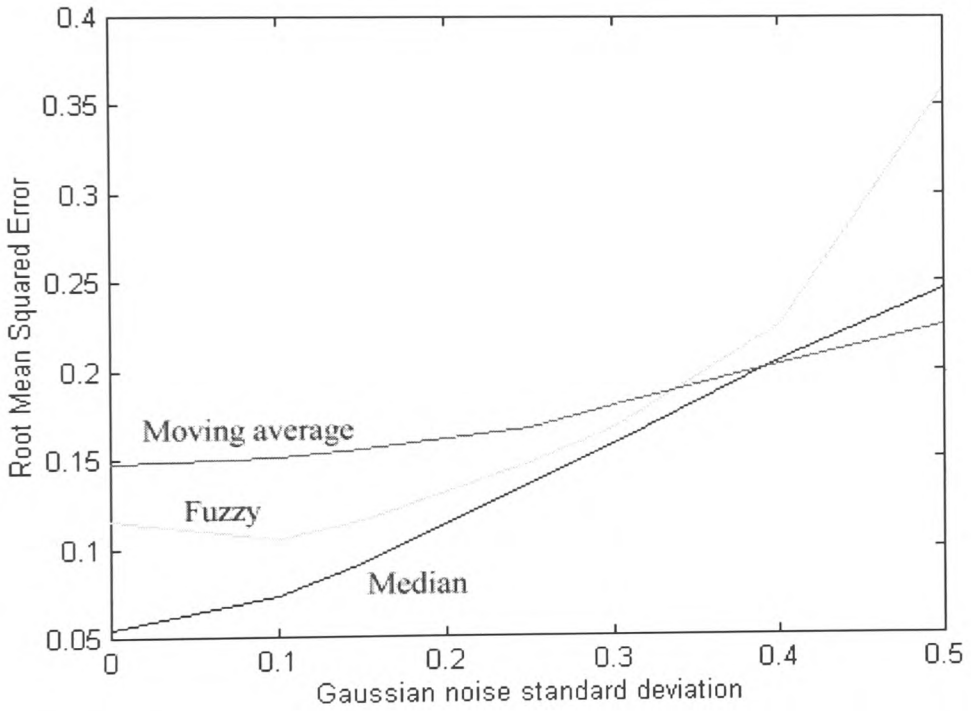

Figure 7.29: Plot of RMSE versus Gaussian noise component of mixed noise for Moving average, Median, and 27 rule fuzzy filter.

It was also observed that the fuzzy filters that gave the best performance were those that were trained to approximate the median. In attempt to improve on the performance of these filters, they were further trained on training data that consisted of signals corrupted with mixed Gaussian and impulsive noise as input data and the true uncorrupted signal as output data. Although in general the training process achieved a low MSE on the training data, in all case

the training process resulted in a filter with poorer performance than the original fuzzy filter. Some of the filters produced a very poor performance on test data. The reasons for this are discussed in section 7.6.4.

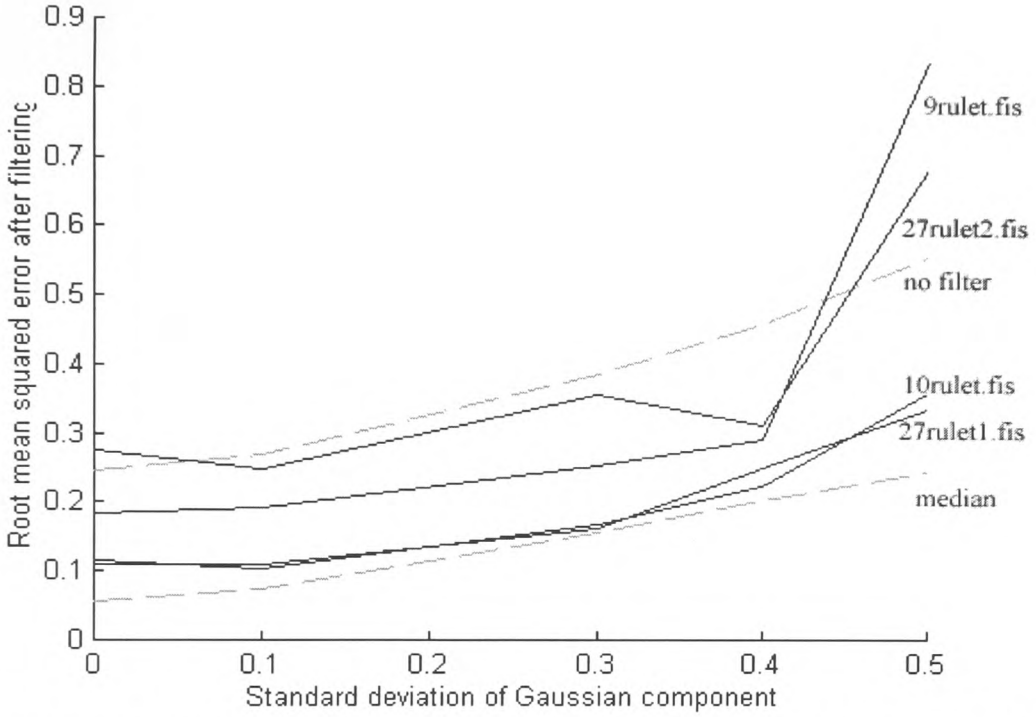

Figure 7.30: Plot of RMSE versus Gaussian noise component of mixed noise before filtering (upper dashed line) after filtering with four different fuzzy filters (solid lines), and after filtering with a median (lower dashed line).

The fuzzy filter with the best performance was also tested by using it to filter the depth maps produced by the WINIM software from the simulated and real sequences of greyscale images 'cake' and 'box'. The result of applying the fuzzy filter, which was a 10-rule approximation to a median, to the depth maps produce by the image sequence 'cake', is shown in figure 7.31. Also shown for comparison are the corresponding results for a 3 x 3 moving average (figure 7.32) and a median filter (figure 7.33). In all cases the Kalman filter was enabled, the variable matching window size was enabled with a maximum of a 9 x 9 window, and the filters were applied between every matching step. The corresponding RMSE at the end of the sequence of five images is also shown in the figures.

Figure 7.34 shows the result of applying the fuzzy filter to the depth map resulting from the real image sequence 'box'. For comparison, the corresponding depth map after applying a 3 x 3 median filter is shown in figure 7.35.

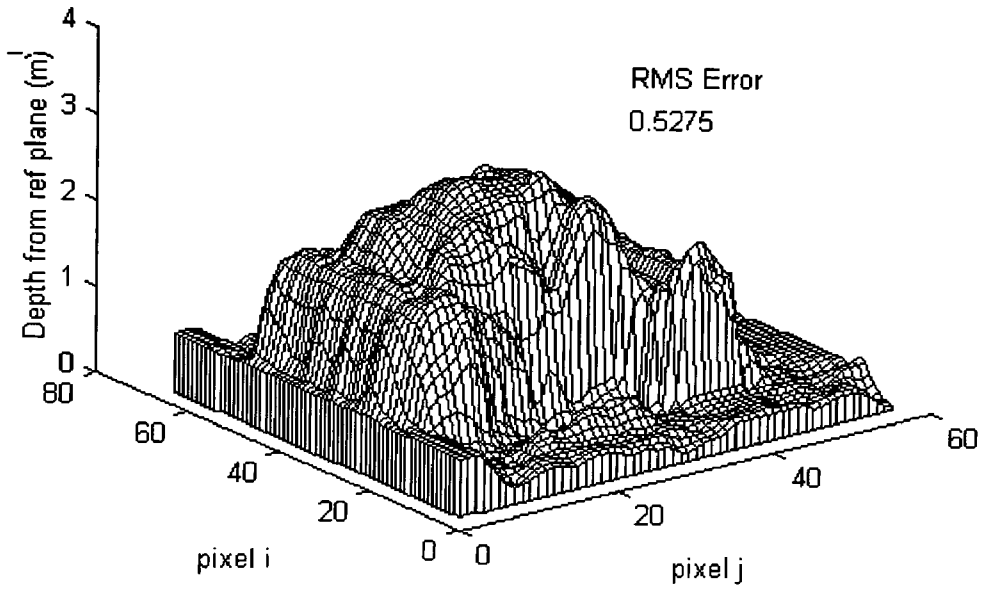

Figure 7.31: The result of applying the fuzzy filter, which was a 10-rule approximation to a median, to the depth maps produced by the image sequence 'cake'.

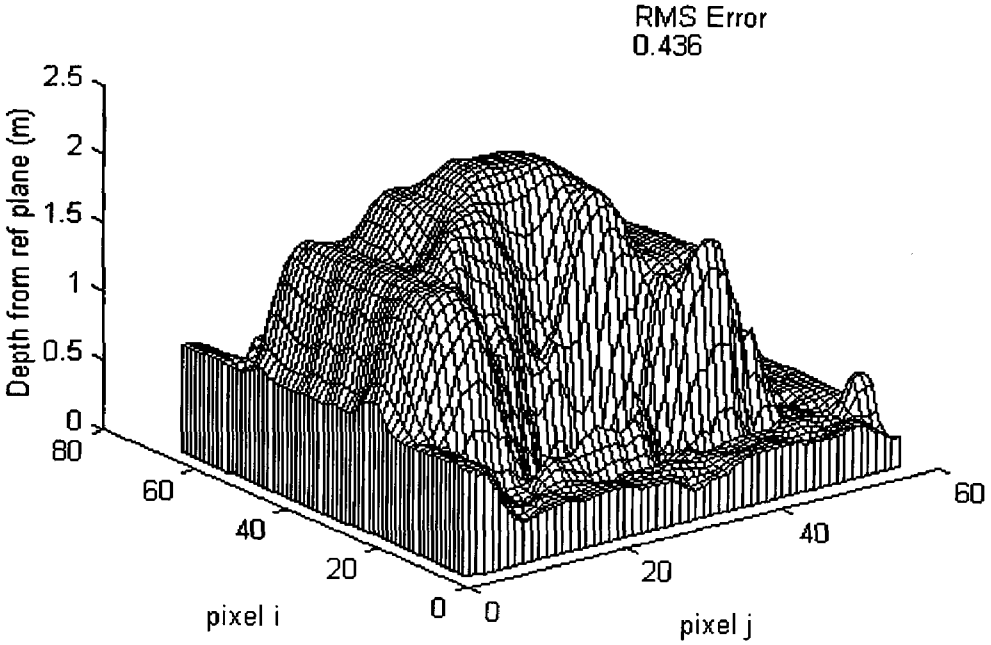

Figure 7.32: The result of applying a 3 x 3 moving average filter to the depth maps produced by the image sequence 'cake'.

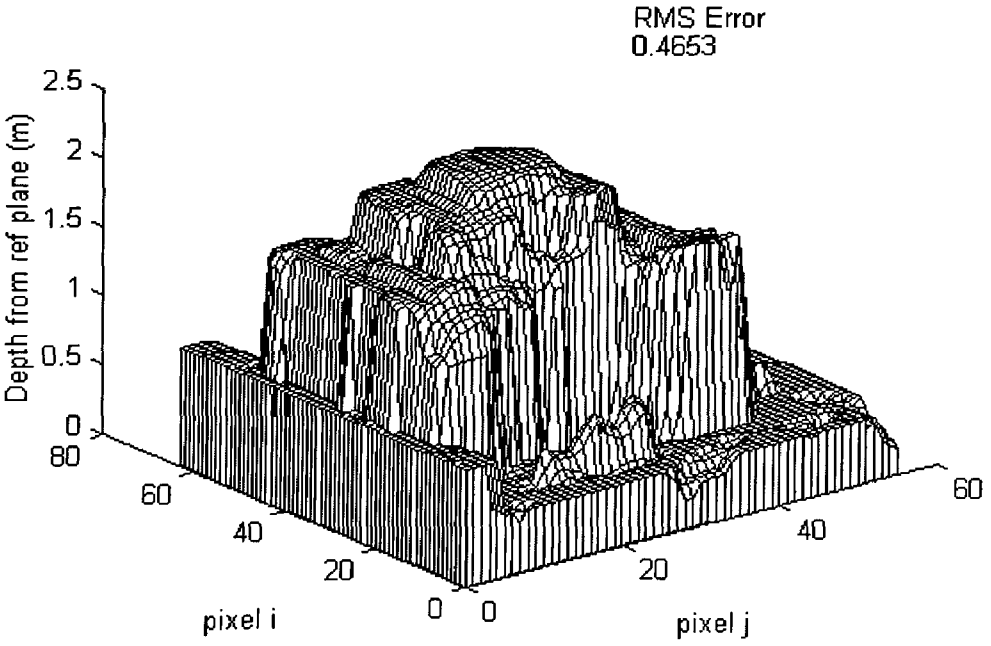

Figure 7.33: The result of applying a 3 x 3 median filter to the depth maps produced by the image sequence 'cake'.

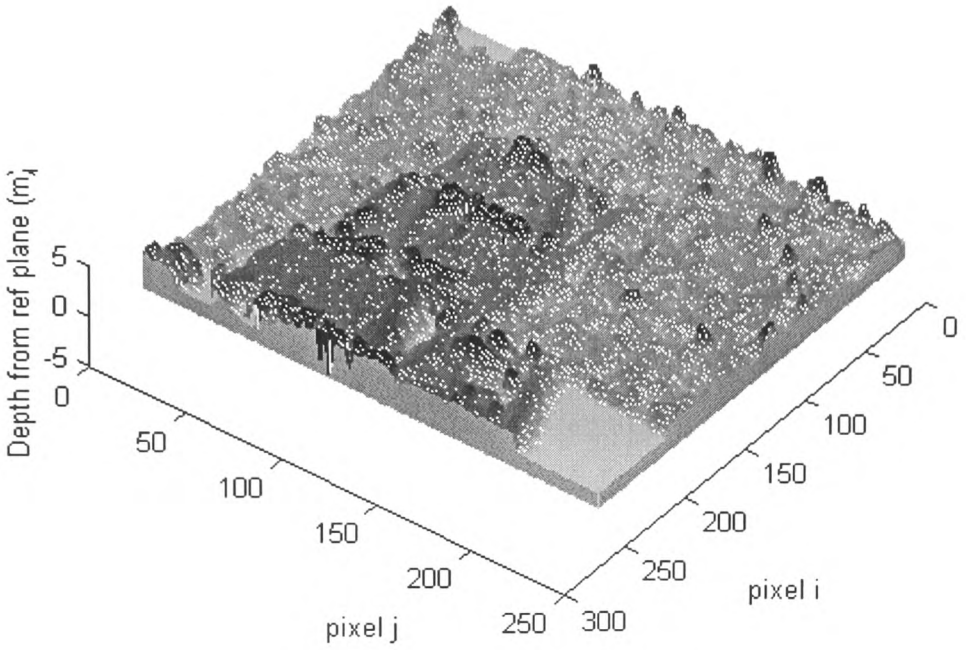

Figure 7.34: The depth map resulting from applying the 10-rule fuzzy filter to the depth map resulting from the real image sequence 'box'.

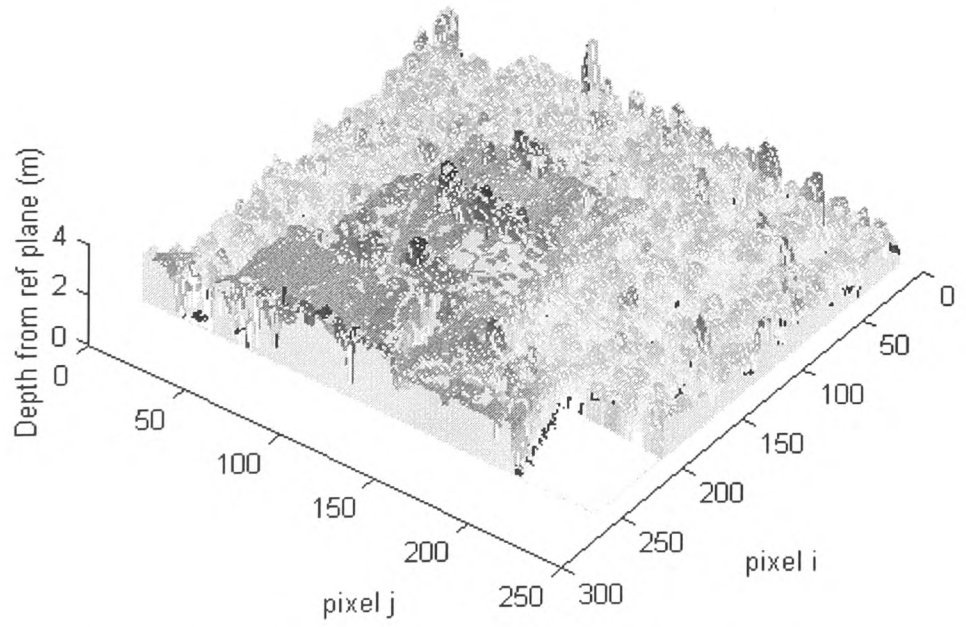

Figure 7.35: The depth map resulting from applying a 3x3 median filter to the depth map resulting from the real image sequence 'box'.

7.6.4 Discussion

One interpretation of the results of section 7.6.3 is that for the mixed noise that has been added to the simulated depth map 'ref', the median filter represents the best possible performance for a given filter size that can be obtained whilst the noise is predominantly impulsive. As the Gaussian noise component increases, the median becomes inferior to the moving average filter. This interpretation relies on the assumption that if the search techniques used to train the fuzzy filters were ideal and if a better nonlinear filter than the median existed, then a fuzzy filter approximating this improved nonlinear filter would have been found. It is, however, difficult to be wholly confident of the effectiveness of the training techniques. It is possible, for example, that the training data set used was insufficiently large or insufficiently representative. This concern is given some support from the observation that some filters behaved pathologically on the depth maps after training despite having a low MSE on the training data. This pathological behaviour took the form of outputting extremely large values for reasonable input vectors where those input vectors did not occur in the training data. In effect this pathological behaviour represents 'overtraining' writ large. A possible avenue for future research is to incorporate a test of the stability to the data of a fuzzy system during training by applying perturbation to the training data set and measuring the change in the value of the output to the perturbation. It is also possible that the number of input fuzzy sets used in the fuzzy systems (three) was inadequate to capture the optimal mapping from noisy input vector to filtered output. Increasing the number of fuzzy sets or the size of the training data set, however, led to unacceptably slow training rates. It is likely that this could be improved by re-writing the FTEST training software, since the current version of this software has been arrived–at by a process of evolution and is almost certainly less than maximally efficient.

7.7 Alternative approach to the use of the Sugeno fuzzy architecture

This section outlines a different approach to the use of a Sugeno type architecture for fuzzy filtering to that taken in the other sections of this chapter. The different approach, which is described in more detail in (Rothwell Hughes *et al.*. 1997) has more in common with the indirect filter approach more commonly taken in the application of fuzzy systems to filtering. It is included here to illustrate that the Sugeno fuzzy filter architecture is flexible enough to contain the strategies used in indirect filters. The filter described here was designed using a heuristic approach and the training techniques described in Chapter 5 were not used.

The input vector to the fuzzy system consists of the nine pixels reading from left to right and down in a 3 x3 filter window; the difference from the filter window median for each of the nine pixels, and the median itself. Thus, the crisp input vector has 19 elements. The differences from median inputs are classified by their membership in three fuzzy sets labelled negative large (NL), positive large (PL) and small (S). The membership functions of these sets are Gaussian in shape and are parameterised by two parameters, width and centre.

The rules for each pixel (l,m) in the filter window having depth $d(l,m)$ are of the form:

IF Difference-From-Median is S THEN output is : $d(l,m)$

IF Difference-From-Median is NL or PL THEN output is : Median.

The output sets are achieved by setting the output set coefficient a_{ji} corresponding to the input values to be one. The effect of this difference-from-median filter (DFM) is to act as a moving average filter in smooth areas and as a median filter in areas containing discontinuities.

The filter was applied between the matching steps during the generation of depth maps from the simulated image sequence 'cake'. After adjustment of the input set parameters the RMSE achieved were as shown in table 7.8.

Filter Type	**RMSE on simulated image**
Moving Average (3x3)	0.2482
Median (3x3)	0.2589
DMF (3x3)	0.2303

Table 7.8 Comparison of RMSE for DFM filter, moving average and median filters.

The Resulting depth map is shown in figure 7.36. It can be seen that there is some loss of definition at the depth discontinuities, although the RMSE is better than either the moving average or the median filters. The discontinuities can be better preserved by adjusting the input set parameters, but only at the cost of worsening RMSE.

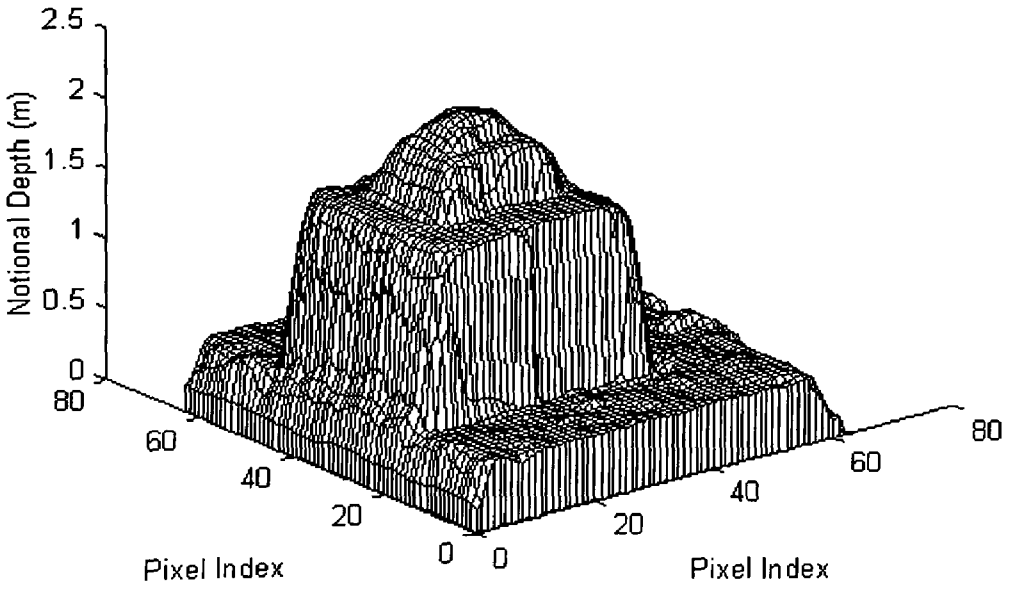

Figure 7.36: Depth map filtered by Fuzzy Difference-From-Median Filter

7.8 Summary and Conclusions

This chapter has briefly described the WINIM software that was used to implement the stereo matching algorithm using the SSD matcher. The software allows the fuzzy systems created and trained in the FTEST software to be applied to the task of filtering the depth maps thus generated. A description was given of the methods used to generate training and test signals corrupted by mixed Gaussian and impulsive noise.

It was shown that the FIR type Sugeno fuzzy filter architecture described in Chapter 6 could be used to approximate a 3 x 3 median filter, which is known to be effective in removing impulsive noise and preserving signal discontinuities. The results of the tests on the effectiveness of various fuzzy systems in matching the training data appear to show that the 'goodness of fit' to the training data improves nonlinearly with the number of rules used. However, this improvement is not always maintained when tested against separate exemplar data.

The FIR type Sugeno fuzzy architecture was then applied to the task of removing impulsive noise from one-dimensional signals. The fuzzy filters generated were trained and tested using mixed noise corrupted test data that was constrained to be very similar in form to the training data. For this type of test data, the fuzzy filters gave a promising performance that compared well with the MSE performance of median, moving average and Wiener filters.

The one-dimensional fuzzy filters were applied to two-dimensional simulated depth maps and depth maps generated from simulated greyscale images. Although the fuzzy filter architecture can be applied to two dimensional filter windows, it is computationally intractable to train the fuzzy systems that result. It was therefore decided to apply the one-dimensional filters along the horizontal and then the vertical directions. The results of doing

this appeared to contradict the results of section 7.5, which suggested that fuzzy filters could approximate filters of a better performance than the median, moving average or even an ideal Wiener filter. The results were suggestive that for noise that was predominantly impulsive and for a fixed size of filter, the median could not be improved-upon. Reasons for why this conclusion might not be valid were also discussed.

Finally, an alternative approach to using the Sugeno fuzzy architecture for depth map filtering was briefly outlined and the results of some tests on depth maps generated from simulated image sequences were presented. This alternative approach is an implementation using the Sugeno fuzzy system of the indirect filter approach in which the fuzzy system arbitrates between two conventional filters. The tests showed that this approach could give results that are an improvement on the median and moving average filters.

7.9 References

(Jang, 1993) J-S R Jang "ANFIS: Adaptive-Network-Based Fuzzy Inference System." IEEE Transactions on Systems Man and Cybernetics Vol 23 No 3 pp 665-684 May/June 1993

(Rothwell Hughes *et al.*, 1997) Rothwell Hughes N., Roberts G.N., Wilson, G.R. "Application of fuzzy signal processing to three dimensional vision" Proceedings of IEE fifth International Conference on Factory 2000 pp 319-324. April 1997.

Chapter 8: Review of thesis, Conclusions, and Further Work

8.1 Review

This thesis begins from the problem of extracting three-dimensional depth information from sequences of two-dimensional images using stereo matching. The SSD matcher is widely used in the literature (Matthies *et al.*, 1989), (Trucco *et al.*,1996), (Anandan, 1984) for stereo matching and a method of extracting sub-pixel measurements of disparity using this matcher has been previously proposed and used (*op cit* Matthies *et al.*, 1989) . The existence of a mixed noise in the disparity map has been noted in (*op cit* Trucco *et al.*,1996). In section 3.3.2, an analysis is made of the noise process associated with the SSD matcher in terms of the statistics of the original greyscale image. The predictions of this analysis on the distribution of the error in disparities are in agreement with the results of tests on random dot images. Further tests presented in section 3.4 illustrate the fact that increasing noise in the greyscale images leads to impulsive noise due to mismatching. The combination of the Gaussian and impulsive noise components leads to a mixed thick-tailed Gaussian-like and impulsive noise observed in the depth maps produced using the SSD matcher.

Depth from stereo using correlation based matching is known to be ill-posed (Bertero *et al.*, 1988) and this results in gross mismatches. These gross mismatches appear as impulsive noise in the resulting depth map. A classical approach, called regularisation, to the resolution of an ill-posed problem is to constrain the solutions to that problem to conform to some prior assumptions (*op cit.* Bertero *et al.*, 1988). These prior assumptions come from some extra knowledge about the form of allowable solutions. Regularisation can be equivalent to a filtering process on noisy data (Terzopoulos 1986(b)). This thesis examines the proposition that filters based on fuzzy inferencing systems offer a flexible method of applying regularisation to the ill-posed problem of disparity and

depth map generation using correlation based matching. The regularising prior assumptions can be coded into the filter by way of the rulebase and the fuzzy system parameters.

A brief review of possible existing techniques that could be used for depth map filtering is made in Chapter 4 and the observation is made that filtering processes can be likened to a function-like mapping process which can be approximated by fuzzy systems. Some of the existing filtering techniques are optimal for certain classes of signal and noise. However in a mixed noise environment no one filtering approach is optimal. The possibility exists for a filter based on fuzzy logic to vary its behaviour under different signal conditions so as to approximate that of an optimal filter for those conditions

Filters based on fuzzy systems inherit the general feature of fuzzy systems that they can be trained using exemplar input and output data. This is referred-to as the neuro-fuzzy approach and following an overview of fuzzy systems in which the Mamdani and Sugeno systems are described, Chapter 5 proposes and investigates six approaches to fuzzy system training based on the simulated annealing algorithm.

It is believed that the interpretation of a fuzzy logic based filter in terms of the ideas of regularisation has not been made before. However, the concept of applying fuzzy logic to filtering problems has been reported in a number of papers. Chapter 6 makes a survey of these papers and proposes a taxonomy of fuzzy logic based filters (fuzzy filters) using the idea of direct and indirect acting fuzzy filters. The majority of filters in the existing literature are of the indirect acting class in which the fuzzy inferencing system is used to control or select the output from a set of conventional filters. Direct acting fuzzy filters encapsulate the whole of the filtering action within the fuzzy system. Chapter 6 gives an example of the application of an indirect acting filter to the smoothing

of disparity maps generated using the SSD matcher. It is believed that there are no previous examples of fuzzy filters being applied to this problem.

The Sugeno fuzzy system architecture lends itself to a filtering application since its output fuzzy sets are functions of the crisp input variables. The FIR type Sugeno filter, introduced in section 5.4, is a special version of the Sugeno first order system in which the constant term is suppressed. Chapter 6 emphasises the suitability of this type of fuzzy system as the basis for a fuzzy filter by drawing a FIR type Sugeno system as a bank of controlled linear filters. The Sugeno system effectively subsumes the mechanism of an indirect fuzzy filter into one fuzzy system, making a direct fuzzy filter structure which can be trained using the techniques discussed in Chapter 5.

Chapter 7 initially considers the problem of generating suitable training data for training Sugeno system based fuzzy filters for the task of depth map regularisation. Using suitable MATLAB script files noisy one-dimensional signals are generated whose histograms mimic those of the disparity and depth maps discussed in Chapter 3. A restricted class of signals can also be generated as the wanted signal.

Fuzzy filters were trained to mimic three element median filters in order to demonstrate that fuzzy filters could approximate this important type of nonlinear filter. The fuzzy filters were then trained on simulated training data to filter a signal corrupted by impulsive noise. After training, most of these filters achieved a mean squared error performance on test data that was better than three element median or moving average filters. Some of the fuzzy filters achieved performances that were better than even an ideal Wiener filter for which the exact signal and noise power spectral density was known. A further fuzzy system was trained to have an error performance that was

better than either a three-element moving average or a three element median filter on test data corrupted by mixed Gaussian and impulsive noise.

Owing to the computational difficulties of training a Sugeno fuzzy filter with a two-dimensional window, one-dimensional fuzzy filters were applied successively along the horizontal and vertical axes. Such filters were applied to simulated depth maps, depth maps generated from simulated greyscale images, and depth maps generated from real greyscale images. Thus far it has proven to be impossible to produce fuzzy filters using this approach that outperform the median filter in the presence of impulsive noise.

Another technique that was discussed and demonstrated in Chapter 7 was to extend the Sugeno system so that some pre-processing of the inputs was allowed which extracted the filter window median and difference from median for each pixel. This technique allows a much simpler rulebase.

8.2 Discussion and Conclusions

8.2.1 Sum of squared difference matcher

The theory of (*op cit* Matthies *et al.*, 1989) suggests that for no greyscale image noise the subpixel SSD matcher should produce zero noise in a disparity map produced using the matcher. However observation of the performance of the SSD sub-pixel matcher on random dot greyscale images suggests that this is not the case and that there is always a minimum, backstop, noise associated with the sub-pixel matcher. An analysis is made in section 3.3.2, of the effect of the greyscale image statistics on this minimum backstop noise. The analysis assumes a simple random uniform distribution of grey-scale values, that the grey scale values are independent, and makes a simplifying approximation that the SSDs on either side of the fitted quadratic minimum are

independent random variables. Nevertheless despite these assumptions, it provides an insight into the disparity map noise that occurs with more complex grey-scale image statistics. For a small 3 x 3 matching window the agreement of the analysis with experiment is good. The poorer agreement for larger matching windows is due to the failure of the assumption of the independence of the SSDs to either side of the quadratic minimum.

The salient conclusions that are drawn from the analysis are that for the cases of independent grey scale pixel values considered, the distribution of disparities is found to be substantially independent of the statistics of the grey scale image in the absence of noise. However, the width of the backstop disparity distribution depends on the matching window size. Larger matching windows give rise to narrower disparity distributions.

Examination in section 3.3.3 of the effect of the greyscale image expansion used in (*op cit* Matthies *et al.*, 1989) shows that the main effect of this image expansion is to reduce the inherent backstop noise of the sub-pixel matcher.

The investigations detailed in section 3.4 confirm the observations of (*op cit* Trucco *et al.*,1996) that the SSD matcher produces impulsive noise due to gross mismatches when additional noise is added to the greyscale images. These gross mismatches would also occur in the presence of image distortion. The disparity maps resulting from the SSD matcher will consist of a mixed impulsive and Gaussian noise. Moreover when depth maps are generated from the disparity maps, the Gaussian noise in the disparity map will be transformed to a different distribution. The transformation will depend on the camera parameters and the value of the disparity. Therefore the distribution of errors in a depth map will not be simply Gaussian noise, but consist of a mixture of impulses and transformed Gaussian noise.

8.2.2 Current filtering approaches

There are many possible linear and nonlinear current approaches that could be applied to depth map filtering. The Wiener filter (Wiener, 1949), as the optimum linear filter can be used as a benchmark where the signal and noise power spectral densities are known. For frontoparallel flat surfaces corrupted by Gaussian noise, a moving average filter can reasonably approximate the Wiener filter. However the average is not stable in the presence of outliers (Huber 1964). The median filter is a nonlinear filter that is robust to outliers and also preserves edges and discontinuities in the signal. For signals such as the mixed noise corrupted depth maps generated by the SSD matcher, a filter which changes its behaviour between the median and the moving average depending on the presence of impulses or discontinuities has the potential to perform better than either a pure moving average or a pure median.

8.2.3 Training of fuzzy systems

In Chapter 5 six different approaches to training a Sugeno fuzzy system were proposed, and tested by testing the ability of the fuzzy system to approximate a simple nonlinear function of two inputs. The simplicity of the function that was chosen to be approximated was such as to allow many training runs to be accomplished in a reasonable time. All the training approaches were based on a combination of known search and optimisation algorithms (Metropolis *et al.*1953) (Nelder and Mead, 1965), (Press *et al.*, 1994) (Jang, 1993). However, although a recent paper has applied an approach that shares certain aspects of the approach taken in this thesis to fuzzy system training (Garibaldi and Ifeachor, 1999) the specific application to the training of Sugeno fuzzy systems is felt to be a new contribution made by this thesis.

The output sets of the Sugeno fuzzy systems, which are defined by the parameter set that scale each crisp input, can be trained by using a linear least squares approach. This approach, used by (Jang,

1993). in conjunction with the backpropagation algorithm, is implemented using an SVD approach to solving the linear least squares problem. The input parameters are trained using different versions of the simulated annealing algorithm

There is a potential problem with the conjunction of the simulated annealing algorithm with the linear least squares algorithm. As implemented in the work described in this thesis all the simulated annealing algorithms are used to minimise the mean squared error whereas the total squared error is minimised by the least squares routine. The target for the simulated annealing algorithms could be changed to the total squared error, but the mean squared error is a more usual measure in noise removing filters, the target application for the fuzzy systems to be trained. This is because the mean squared error is associated with the power of a zero mean noise process. Moreover improvement in the total squared error results in an improvement in the mean squared error. A practical problem that has been encountered is that the numerical routine (taken from (Press *et al.*, 1994)) used to perform the SVD as a step in the least squares algorithm, often fails to converge for large matrices.

After training of a Sugeno system using the training methods described in Chapter 5, the supposed advantage of the transparency of fuzzy systems disappears unless steps are taken to preserve that transparency. These steps are not taken in the training routines used in this thesis, and therefore since training using exemplar input-output data plays a central role in the approach to fuzzy filter design taken in this thesis, the resulting fuzzy systems are not amenable to having their action understood by examination of the fuzzy system parameters.

8.2.4 Direct and indirect acting fuzzy filtering systems

A review of the literature covering the topic of filters based on fuzzy logic reveals that the majority of the published work on fuzzy filters concerns the class of indirect filters. In this type of filter the fuzzy system is used as a control mechanism that arbitrates between several conventional filters. Direct filters, in which the data enters the fuzzy system and the outputs emerge directly from the outputs of the fuzzy system alone are very rare. Direct acting filters have an inherently high dimensionality (many inputs), which brings with it the problems of rulebase explosion, a large search space volume for training, and the need for a large training data set. Nevertheless fuzzy systems are capable of universal approximation (Wang, 1992), (Kosko, 1992) which make them into candidates for filtering tasks. Also the first order Sugeno fuzzy system (Takagi and Sugeno, 1985) can approximate complex mappings with fewer rules than the Mamdani model.

8.2.5 Filters implemented using a Sugeno fuzzy system

The first order Sugeno fuzzy system can be drawn as a feedforward filter network. The mapping performed by this network is nonlinear, even though the output sets are linear combinations of the inputs. The filter structure that results is sensitive to input level, not only because of the nonlinear nature of the filter, but also because the filter is designed to work over a finite universe of discourse. Because of the level sensitivity, it is better to apply the fuzzy filter to depth rather than disparity map filtering when the disparities are derived from sequences of greyscale images. This is because the range of depths encountered is constant at every matching step, whereas the disparity evolves through the sequence.

The Sugeno network can approximate a median function with three inputs. It can also approximate a five-input median. An exhaustive rulebase is not necessary to implement the median filter. There appears to be a strong relationship between the number of rules in a rulebase and the ability of a

fuzzy system to match training data. However, the relationship between the number of rules and the ability to generalise to the underlying model behind the training data does not appear to be so strong, at least for a three input median function.

It is possible for a Sugeno filter trained to filter noise on a very restricted type of signal to outperform median, moving average and even Wiener filters when applied to a different signal that is restricted in the same way as the training data. However, it is possible that for this case the Sugeno filter has learnt the salient aspects of the signal, rather than the function necessary to filter noise from that signal.

The extension of Sugeno-based filters to two-dimensional filters causes computational problems if it is necessary to train the resulting filters. This is because of the dimensionality problems inherent in fuzzy systems. Non-exhaustive rulebases where the number of rules used as a fraction of the exhaustive number is small have not yielded good results when used as filters, despite using the rule selecting training method described in section 5.11. This may be because the training data used was inadequate, the search methods ineffective, or because the small number of rules were insufficient to capture any useful mapping from the training data.

When Sugeno based fuzzy filters were applied to mixed noise corrupted two-dimensional signals by filtering successively in the horizontal and vertical directions, those one-dimensional fuzzy filters that were designed to operate as median approximators produced the best results. These results were never better, however than those for a median filter which delineated a lower bound on the MSE that was achieved by any of the fuzzy filters. Some filters that were trained on mixed noise signals produced very large incorrect outputs when presented with data that did not appear in the training data set, even though the error achieved over the training data set was reasonable. This

suggests that the training methods should be modified to ensure that the fuzzy filter is not overly sensitive to the input data

The results observed on a simulated depth map are reflected in the performance of the fuzzy filters on depth maps produced from simulated image sequences, except that for these depth maps the fuzzy filters produced worse results than both the median and moving average filters. When the filters have been applied to depth maps derived from real image sequences it has not been possible to make direct comparisons of MSE between the different filters since the ground-truth depth map is not known. It is apparent, however, that with the matching techniques used in WINIM the filtering problems are much more severe than those encountered with the simpler more idealised simulated images.

8.3 Further Work

The matching routines as currently implemented in WINIM are crude. In particular the geometrical model used for the perspective projection is the simplest possible. Since the focus of interest of the thesis was in the application of fuzzy systems to filtering, the matching routines were not developed to eliminate the errors that were introduced by the simplicity of the model. Instead much of the work has been carried out on simulated images that matched the assumptions of WINIM. The simulated image sequence also has the advantage of having a known but simple depth map which has only frontoparallel surfaces. It would be useful and interesting to implement a more sophisticated matching model and concurrently to verify this model on more sophisticated simulated imagery. This simulated imagery would in effect be a 'simulated calibrated imaging laboratory'. Using this model the distortions that lead to mismatching could be closely examined and hence a better statistical model of the disparity map noise developed.

The efficiencies of the training routines described in Chapter 5 should be examined so that the dimensionality problem inherent in fuzzy system training is not exacerbated by an inefficient implementation. The annealing schedule used in the simulated annealing routine could also be modified so as to be controlled by the rate of change of error.

The loss of transparency of the fuzzy system after training negates the salient advantage of fuzzy systems over neural networks. It would be useful to restrict the changes that the training routine can carry out so that some of the transparency of the fuzzy system is not lost. An example of this would be to prevent input sets from being reordered during training.

Two critical questions for fuzzy system training are the question of the sufficiency of training data, and the stability of a fuzzy system's output to small changes in input data. It would be useful if this stability could be assessed during the training of a fuzzy system by factoring a measure of the stability into the cost function evaluated by the simulated annealing routine. This could be achieved by perturbing the training data by a small amount and assessing the change in error rate. Alternatively a second training data set could be used instead of a perturbation to assess the sensitivity of the error rate to the data. The size of the training data set is a thorny question since too small a data set may lead to the problems of a fuzzy system being unable to generalise, whereas too large a training data set slows the training process. The relationship between the necessary size of the training data set and the task to which the fuzzy system is to be applied represents a large field of potential future research.

The fundamental task which the work described in this thesis has attempted was to establish whether some nonlinear filter which could be approximated by a fuzzy system could improve on the performance of existing conventional filters when applied to the task of depth map filtering. This

target remains as a challenge to be achieved, but the computational challenges of a search that scales as the power of the input vector length are profound. Given the problems of fuzzy system dimensionality, the immediate future of research into the application of fuzzy systems to filtering may lie with more research into the design and behaviour of one-dimensional fuzzy filters.

8.4 References

(Anandan, 1984) Anandan P. "Computing dense displacement fields with confidence measures in scenes containing occlusion. " Proceedings DARPA image understanding workshop. pp 236-246, 1984.

(Bertero *et al.*, 1988) Bertero M, Poggio T and Torre V, "Ill Posed Problems in Early Vision", Proceedings IEEE , 76 , pp869 – 889 1988 .

(Garibaldi and Ifeachor, 1999) Garibaldi J. M. and Ifeachor E.C. "Application of Simulated Annealing Fuzzy Model Tuning to Umbilical Cord Acid-Base Interpretation." IEEE Transactions on Fuzzy Systems Vol 7. No 1 pp 72- 84. Feb 1999.

(Huber 1964) Huber P.J. "Robust estimation of a location parameter." Annals of Mathematical Statistics Vol 35 pp73-104 1964.

(Jang, 1993) J-S R Jang "ANFIS: Adaptive-Network-Based Fuzzy Inference System." IEEE Transactions on Systems Man and Cybernetics Vol 23 No 3 pp 665-684 May/June 1993

(Kosko, 1992) Kosko B. "Fuzzy Systems as Universal Approximators." IEEE International Conference on Fuzzy Systems San Diego. pp 1153-1162. 1992.

(Matthies *et al.*., 1989) Matthies L., Kanade T.,and Szeliski R., "Kalman Filter-based Algorithms for Estimating Depth from Image Sequences." International Journal of Computer Vision. No3, pp 209-236, 1989.

(Metropolis *et al.*.1953) Metropolis N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H., Teller E. "Equation of State Calculations by Fast Computing Machines" Journal of Chemical Physics Vol 21, No 6. pp 1087-1092. June 1953.

(Nelder and Mead, 1965) Nelder J.A, and Mead R. "A simplex method for function minimization" Computing Journal. Vol 7 pp 308-313 1965.

(Press *et al.*, 1994) Press W.H., Teukolsky S.A., Vetterling W.T., and Flannery B.P. "Numerical Recipes in C The Art of Scientific Computing"2nd Edition C.U.P. 1994.

(Terzopoulos 1986(b)) Terzopoulos D. "Regularisation of Inverse Visual Problems Involving Discontinuities." IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 8 No. 4 pp 413 - 424, July 1986.

(Trucco *et al.*,1996) Trucco E, Roberto V, Tinonin S and Corbatto M. "SSD Disparity Estimation for Dynamic Stereo", Proceedings British Machine Vision Conference Vol. 1, pp 343-352. 1996.

(Wang, 1992) Wang L-X. "Fuzzy Systems are Universal Approximators." IEEE International Conference on Fuzzy Systems San Diego pp 1163-1169. 1992

(Wiener, 1949) Wiener N. "Extrapolation, Interpolation, and Smoothing of Stationary Time Series, with Engineering Applications." MIT Press 1949.

(Takagi and Sugeno, 1985) Takagi T and Sugeno M, "Fuzzy identification of systems and its applications to modelling and control." IEEE Transactions on Systems Man and Cybernetics Vol 15 pp 116-132 1985.

Appendix A: Papers

Appendix A contains copies of all the papers that have been produced and published from the research described in this thesis. The papers are:

Rothwell Hughes N., Wilson G. R., and Roberts G. N. "Kalman and Fuzzy Logic Filters for 3-D Industrial Inspection Using a Single Camera." Proceedings of the International Conference on Recent Advances in Mechatronics (ICRAM95). pp 595-600 August 1995.

Rothwell Hughes N., Wilson G., and Roberts G., "Fuzzy Regularisation of Depth Maps Derived from Image Sequences", International Conference on Mechatronics and Machine Vision in Practice Vol 2 pp 55-60. 1996.

Rothwell Hughes N., Roberts G.N., Wilson, G.R. "Application of fuzzy signal processing to three dimensional vision" Proceedings of IEE fifth International Conference on Factory 2000 pp 319-324. April 1997.

Kalman and Fuzzy Logic Filters for 3-D Industrial Inspection Using a Single Camera

N Rothwell Hughes, G R Wilson, G N Roberts.
Gwent College of Higher Education, PO Box 180, Newport, Gwent, NP9 5XR, U.K. email: nrhughes@gwent.ac.uk

Abstract The work described here uses a sequence of images taken at closely spaced intervals with a camera moving in a known way to extract a dense depth map of the viewed scene. The small step size between images, and assumptions about the range of depths in the viewed scene limit the search space in the image pairs for a stereo matching algorithm. A scalar Kalman filter is used to track the increasing pixel shift or disparity between corresponding points in the images. The resulting disparity map can be converted into a depth map using the inverse perspective projection. The disparity map obtained using the above technique is contaminated by noise resulting from gross errors in the correlation. This necessitates a smoothing stage in the algorithm which is based on a priori assumptions about the true disparity map and the corresponding underlying depth map. A filter based on fuzzy logic is used to smooth the disparity map.

1. INTRODUCTION

The ability to use machine vision to produce a 3 dimensional map of a part of the real world is needed both for 3-D industrial inspection and for robot or vehicle guidance. The work described here is directed towards that end. The approach taken uses a single CCD camera under known motion to generate a depth map. This approach is based on a technique which has been described by other workers [1],[2]. In contrast to the previous work, however, a filter based on fuzzy logic is used to smooth the resulting output.

The motion of the camera is a sequence of small steps at each of which an image is taken. By measuring the shift in the image, or disparity, of corresponding points between the first and subsequent images the depth of those points can be determined using the perspective projection. Because the camera step size is small, and assuming a range of depths which are of interest, the search space for matching between the images can also be made small. If the camera motion is also constrained to be in the same plane and parallel to its scanline axis for all steps the search space is confined to a search along a scanline (the epipolar constraint) [3]. By tracking the disparity through a sequence of small steps, a wide baseline can be built up and the precision in the depth map made as great as is required.
The matching is carried out using the sum of squared difference technique (SSD) [4]. A scalar Kalman filter is used to track the increasing disparity for each pixel as the image sequence progresses. After the matching process for an image pair and after updating the disparity map with the Kalman filter, a noisy disparity map is obtained. It is at this stage that the Fuzzy filter is applied. Fuzzy filters have been used in image restoration [5] and their nonlinear filtering properties are ideal for smoothing a disparity or inverse depth map, whilst preserving the edges or areas of high disparity gradient.

2 ALGORITHM TO GENERATE DISPARITY MAP

The stages of the algorithm used to generate the disparity map are as follows:

(i) The first and second intensity images from the image sequence are read in as 256 X 256 matrices, each entry in the matrices corresponding to a gray scale value in the images

(ii) The Sum of Squared Differences (SSD) centred on each pixel (i,j) in image 1 and a 3 X 3 patch centred on pixels (i,j+shift) in image 2 are computed for shifts from -1 pixel to +5 pixels. The SSD is a simple measure of match between the two patches.

(iii) The shift which generates the minimum SSD, 'best_shift', and its two neighbours 'best_shift ± 1' are stored along with their SSDs.

(iv) A quadratic is fitted to these three points. The minimum of the quadratic gives a sub pixel estimate of the shift or disparity for pixel (i,j). The width of the quadratic function (i.e. the coefficient of the square term) is taken as a measure of the uncertainty in the disparity measurement (*op cit*[1]).

(v) Steps (ii) to (iv) are then iterated with increasing matching patch size, until the uncertainty measure has dropped below a threshold and ceased to improve, or the matching patch exceeds

a maximum size. The rate of increase of the patch size is controlled by the incremental improvement in the uncertainty.

(vi) The disparity with least uncertainty and the uncertainty measure are passed to a scalar Kalman filter. The filter tracks the evolution of the disparity for each pixel from image 1 to n which is modelled as a linear function of the camera shifts 1 to n. The prior disparity is assumed to be zero for all pixels

(vii) After the above steps are carried out for all pixels a disparity map, or inverse depth map, exists for all pixels. This map is then passed to a fuzzy logic filtering stage for smoothing. The final smoothed map is stored.

(viii) All the above steps are repeated with images 1 and n where n= 3,4,5...
The search space along the scanline within which matches are sought for each pixel is determined by the overall uncertainty in the disparity for that pixel. Pixels with a large uncertainty in disparity having a larger search window. The search window is centred on the predicted disparity based on the gradient of disparity with camera shift. At each stage the Kalman filter updates its estimate, and the fuzzy smoother smooths the disparity map.

(ix) At the end of a sequence of images a disparity value has been obtained for each pixel in the original intensity image. Assuming the perspective projection and known camera motion, a dense x,y,z map of the original scene can be recovered.

3 KALMAN FILTER

The scalar Kalman filter [6] is used in the algorithm to track the evolution of the disparity value for each pixel(i,j). The usual equations for the Kalman filter become in this case:

Process Model:

$$d_{[n+1]} = \frac{(n+1)}{n} \cdot d_{[n]} + g_{[n+1]} \qquad (1)$$

Measurement Model:

$$M_{[n+1]} = C.d_{[n+1]} + v_{[n+1]} \qquad (2)$$

Estimator Equation:

$$\hat{d}_{[n+1]} = \frac{(n+1)}{n} \cdot \hat{d}_{[n]} + K_{[n+1]} \cdot [M_{[n+1]} - C.\frac{(n+1)}{n} \cdot \hat{d}_{[n]}] \qquad (3)$$

Updated Mean square error:

$$P_{[n+1]} = \frac{1}{C} \cdot K_{[n+1]} \cdot \sigma^2_v \qquad (4)$$

Filter Gain:

$$K_{[n+1]} = \frac{C\,[(\frac{n+1}{n})^2 \cdot P_{[n]} + \sigma_g^2]}{\sigma_v^2 + C^2.\sigma_g^2 + C^2.(\frac{n+1}{n})^2 \cdot P_{[n]}} \qquad (5)$$

Where $d_{[n]}$ = disparity in pixels at camera step (image) n; $g_{[n]}$ is zero mean Gaussian noise variance σ_g; $M_{[n]}$ = measured disparity; C = measurement constant=1; $v_{[n]}$ is zero mean Gaussian variance σ_v; $\hat{d}_{[n]}$ = estimated disparity at step[n];

4. DISPARITY MAP SMOOTHING

The general problem of extracting depth from pairs of 2-D images using SSD matching is an example of an ill posed inverse problem [7] In order to solve this problem it is necessary to impose constraints on the allowable solutions so as to choose the best solution given uncertain and ambiguous measurements. In the approach discussed here the small step size between images, and the bounding of scene depths form some additional a priori constraints on the possible range of solutions. The Kalman filter also imposes a constraint on the allowable solutions, given the latest and past disparity measurements, and a measure of the variance of those measurements. Nevertheless, the disparity map which is obtained is noisy, and contains areas of gross error where the matching algorithm fails. A general approach to such problems is that of regularisation. In this approach the optimum solution is that which jointly minimises two energy or cost functionals. The first cost functional measures the closeness of a solution to the measurement, whilst the second measures how well the solution satisfies some a priori constraint such as smoothness. The second cost functional is often called the regularising operator or stabilising functional. A typical pair of cost functionals for the problem being considered here might be:

$$E_1 = \int\int [\tilde{d}(x,y) - d(x,y)]^2 dxdy \qquad (6)$$

and

$$E_2 = \int\int [\nabla d(x,y)]^2 dxdy \qquad (7)$$

where d is the disparity estimate, $\tilde{d}$ is the measurement, and ∇d is the gradient of the disparity estimate, x,y are the co-ordinates in the image, and the integrals are evaluated over the image area.

so that the overall functional E_t to be minimised becomes:

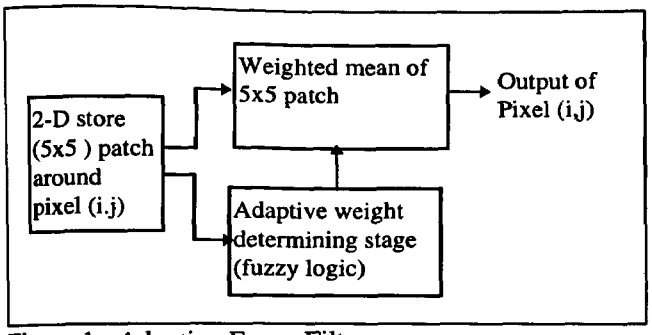

Figure 1: Adaptive Fuzzy Filter

$$E_t = E_1 + \lambda E_2 \quad (8)$$

The exact choice of the stabilising functional and the parameter λ is a choice of a priori beliefs about the problem, and of confidence in those beliefs relative to the data. The minimisation of the functional is often carried out using iterative gradient descent methods and can require much computing time. The process of minimising the energy functional can be thought of as a processing module which has as its inputs the measured disparities and the a priori beliefs about reasonable constraints on the smoothness of the disparity map; and has as its output the disparity map solution which best fits both the measured disparities and the smoothness constraints. A fuzzy logic system can perform the task of such a processing block, where the a priori assumptions are embodied in the fuzzy rule base and the input and output membership functions.

5. FUZZY FILTER

Fuzzy logic was first described as a technique for decision making in the presence of uncertainty by Zadeh [8]. Since then its use in control engineering has grown because of its effectivenes in controlling complex and non-linear systems. A particular advantage is the use which the designer of a system using fuzzy logic can make of 'expert' knowledge expressed in linguistic terms. A fuzzy system maps input values to output values in a way which is determined by a set of linguistically expressed rules. This mapping can be complex and non-linear.

A fuzzy logic system is generally broken down into four parts. The first part, the fuzzification stage takes crisp numerical values and determines their degree of membership in each of a collection of sets which are given linguistic labels that are meaningful in terms of the problem to be solved. e.g. near, local, far. The second part is the fuzzy rule base which expresses relations between the input fuzzy sets A and B and the output fuzzy sets C in the form of 'IF A and B THEN C'. The fuzzified inputs are combined using these rules in the third part, the fuzzy inference engine. This produces a combined fuzzy output set. The final part, the defuzzifier, produces a crisp output from the combined fuzzy output set, usually by taking the centroid of the combined output set.

The fuzzy filter is structured in a similar way to a 2-D finite impulse response filter (figure1), but the filter weights applied to each pixel in a 5 X 5 filter window are determined by the output of a fuzzy logic system. This approach was used by Taguchi et al *op cit* [5] for restoration of grayscale images.

The crisp input values that are fuzzified are:

(i) the euclidean distance of a pixel whose weight is being evaluated, Pixel(i+h,j+k) from the pixel to which the smoothed disparity output will be assigned, Pixel(i,j). This distance is:

$$\text{pixel dist} = \sqrt{h^2 + k^2} \quad (9)$$

and

(ii) a measure of the difference in disparity of pixel (i+h,j+k) from the median of the disparities in the 5X5 filter window, d_median. This disparity difference measure is:

$$\left| \frac{d(i+h,j+k) - d_{median}}{\text{divisor}} \right| \quad (10)$$

where divisor = d_{median} when $d_{median} > 1$,
and divisor = 1 otherwise.

The disparity measure is scaled to the local median disparity so that the filter has the same relative smoothing effect as the disparity measures evolve through the image sequence.

The three fuzzy input sets for pixel distance are labelled Near, Far, and Local. They have membership functions as shown in figure 2.

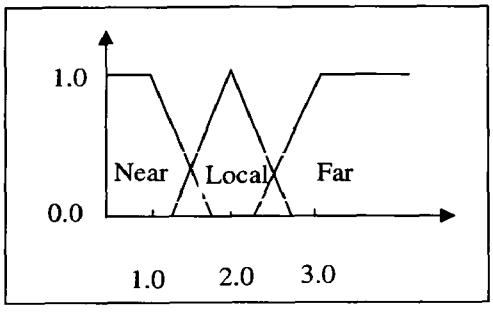

Figure 2: Pixel distance membership functions.

The three sets for disparity difference are Small, Medium, and Large with triangular/trapezoidal membership functions and are shown in figure 3. The shape of the membership functions

are chosen to be triangular/trapezoidal for ease of computation. They need not in general be symmetrical.

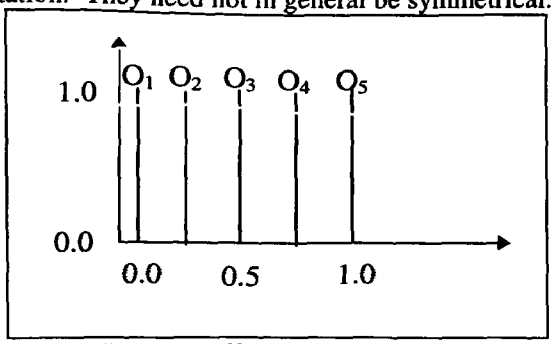

Figure 3: disparity difference membership functions.

Consider the following 5X5 window to be filtered:

4 3 4 5 3	
2 4 2 3 2	
4 7 5 4 5	The window
4 5 6 5 4	median = 4
5 4 4 4 3	

The centre disparity value is the one to be filtered. The underlined disparity is at a distance 1 pixel from the centre pixel, and so its membership values in the distance sets are:

$$\{M_{near}(1), M_{local}(1)M_{far}(1)\} = \{1\ ,0\ ,0\ \} \qquad (11)$$

The disparity difference is:

$$\text{disparity difference} = \frac{7 - 4}{4} = 0.75 \qquad (12)$$

with membership values:

$$\{M_{small}(1), M_{medium}(1)M_{large}(1)\} = \{0.0\ ,0.33\ ,0.33\ \} \qquad (13)$$

The fuzzy rule base and inference engine combine the antecedent fuzzy sets two at a time and associate them with an output or consequent set. The fuzzy rule base used is shown in figure 4.

	Small	Med	Large
Near	O_5	O_3	O_2
Local	O_4	O_3	O_1
Far	O_3	O_2	O_1

Figure 4. Fuzzy Rule base for filter.

The output or consequent sets are fuzzy singletons in this case in the interests of computational simplicity. The membership functions for the output sets are shown in figure 5. Their labels are: O_i , i=1..5

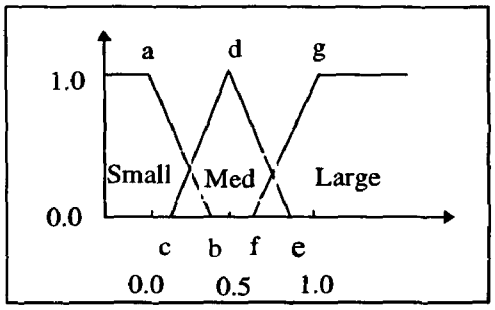

Figure 5: Output fuzzy sets (singletons).

The rule base reflects the a priori assumptions about the correct form of the disparity map; pixels are correlated in proportion to their separation and pixels of markedly different disparity belong to different surfaces. The first rule in the rule base can be read as:

IF pixel distance is Near AND disparity difference is Small Then Output is O_5

For this rule a weight w_1 is assigned to the consequent fuzzy singleton O_5. The weight is determined by taking the minimum of the antecedent membership function values. i.e.

$$w_1 = \min[M_{near}(1), M_{small}(0.75)] = \min[1\ ,0\ ] = 0 \qquad (14)$$

This is repeated for all the rules in the rule base, until a weight w_i is associated with each rule R_i and its consequent fuzzy singleton O_i. The final crisp output, O(h,k) is computed by the weighted sum over all the N rules:

$$O(h,k) = \sum_{i=1}^{N} \frac{w_i O_i}{w_i} \qquad (15)$$

This crisp output is now taken as the weight of pixel (i+h,j+k). The process of obtaining filter weights is repeated for all the pixels in the 5X5 filter window. The final filtered output which replaces the unfiltered value of disparity for pixel (i,j) is calculated from:

$$d_{fil}(i,j) = \frac{\sum_{h=-2}^{h=+2}\sum_{k=-2}^{k=+2} O(h,k)d(i+h,j+k))}{\sum_{h=-2}^{h=+2}\sum_{k=-2}^{k=+2} O(h,k)} \quad (16)$$

6 RESULTS

The algorithm was first tested on some simulated image data. This simulated data consisted of a sequence of ten random dot pattern images generated using MATLAB (figure 6). The images consisted of a fixed random dot background and a square area of a different random pattern which was shifted by 2 pixels between each image in the sequence. The moving foreground is highlighted in figure 6 by the white box, which was not present in the original images. The shifting foreground was allowed to occlude and dissocclude the background as it shifted.

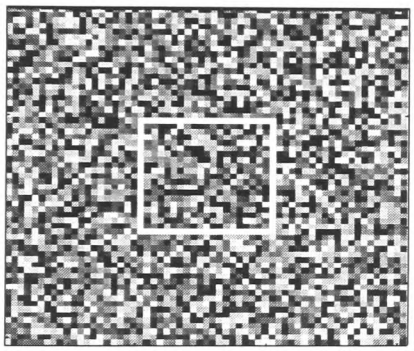

Figure 6: Part of first image of simulated image sequence.

The algorithm was first tested using this simulated data without the fuzzy filter, and the resulting disparity map is shown in figure 7

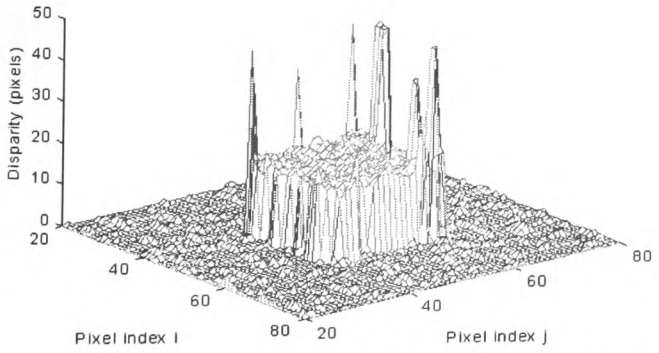

Figure 7: Disparity map without filter,simulated images.

The algorithm was then tested again on the simulated data, but with the fuzzy filter applied to the disparity map between the processing of each image. The parameters of the disparity difference fuzzy set membership functions were then adjusted and the algorithm tested again. The performance of the fuzzy filter was assessed each time by measuring the mean squared error between the ideal disparity map for the simulated data, which is illustrated in figure 8 and the actual disparity map obtained using the algorithm with the fuzzy filter (figure 9). In this way the parameters a,b,c,d,e,f,g of the input fuzzy set membership function (figure 3) were adjusted for lowest mean squared error.

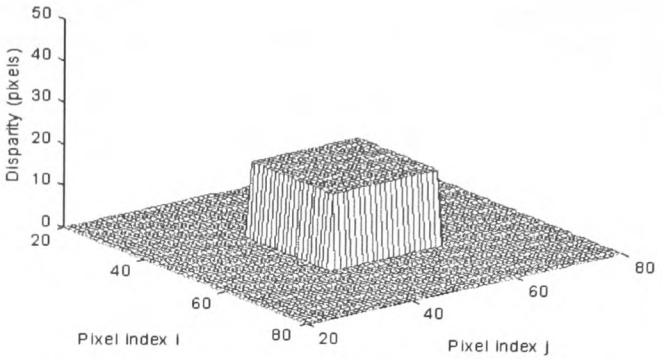

Figure 8: Ideal disparity map for simulated images.

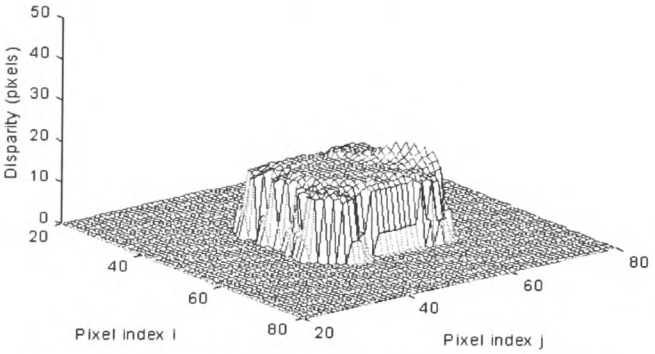

Figure 9: Disparity map with filter, simulated images.

The algorithm was then tested on some real image sequences using the fuzzy set parameters obtained by adjusting on the simulated images. These image sequences were obtained from a CCD camera which was moved in steps of 3mm to a precision of $\pm 10^{-5}$ m. The focal length of the lens was about 25mm and this gave a shift between images of about 2 pixels for objects at 1m from the camera, which is the distance of the nearest part of the first image in the sequence shown in figure 10.

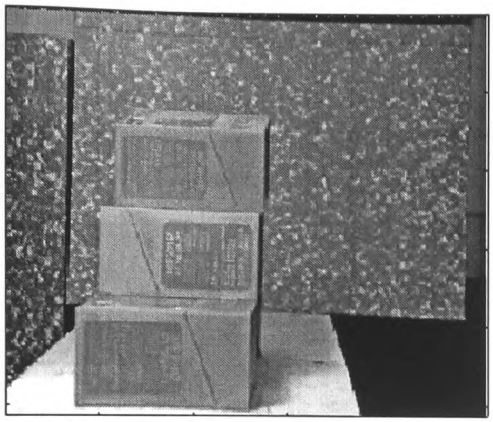

Figure 10: First image of sequence.

The disparity map which was obtained without the fuzzy filter is shown in figure 11 and that obtained with the fuzzy filter is shown in figure 12.

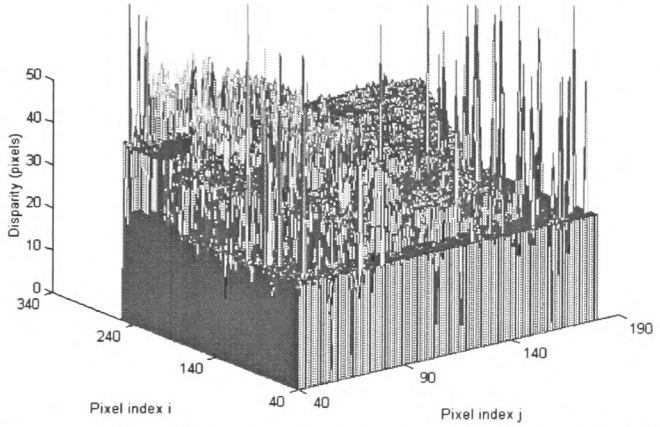

Figure 11: Disparity map for image in figure 10 generated after 20 camera shifts without fuzzy filter.

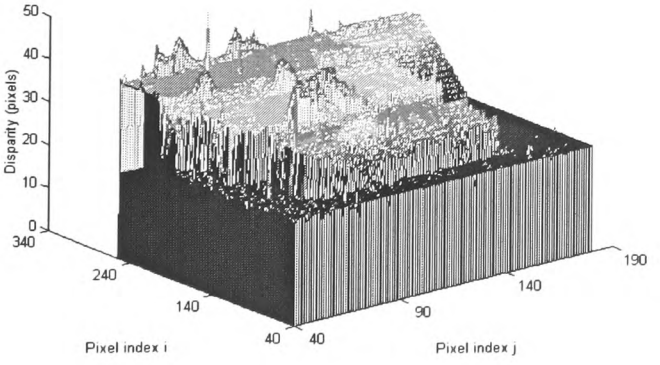

Figure 12. Disparity map for image in figure 10 generated after 20 camera shifts with fuzzy filter.

It can be seen that the fuzzy filter improves the performance of the algorithm. The disparity map that is obtained reflects the depth map for figure10.

7 CONCLUSIONS AND RECOMMENDATIONS

This paper has presented work aimed at generating dense depth maps of a viewed scene using a sequence of images taken by a camera moving in small steps. As in previous work a Kalman filter is used to track the evolving disparity. However the resulting disparity map contains gross errors in some areas due to mismatching in the correlation stage. The Kalman filter does not correct these errors since it only tracks one single pixel, and areas in the images which cause mismatching for one pair of images tend to do so for all pairs of images. Because of this a method of smoothing the disparity map is needed. In the work described here the smoother is interleaved with the matching and Kalman filter stages. The smoother used is an adaptive filter operating over a 5X5 patch in the disparity map. The filter weights are adaptively determined using fuzzy logic. The results on simulated and real data show that such a fuzzy logic based filter can be effective in smoothing noisy disparity maps.

The results gained thus far show that filters based on fuzzy logic are useful in obtaining a solution to the inverse problem of obtaining depth from sequences of images. The use of fuzzy logic provides a framework for incorporating a priori assumptions about the underlying depth map in the solution.

REFERENCES

1. MATTHIES L., KANADE T., SZELISKI R., "Kalman Filter-based Algorithms for Estimating Depth from Image Sequences." International Journal of Computer Vision , 209-236, No3 1989.

2. DISTANTE A., LOVERGINE F.P., ATTOLICCO G., CHIARADIA M. T., CAPONETTI L. "Stochastic structure Estimation by Motion." , Proceedings SPIE Vol. 1826 Intelligent Robots and Computer Vision XI , 476-485 1992.

3. HARRALICK R., SHAPIRO L., Computer and Robot Vision Vol II, 357-359, Addison Wesley, 1993.

4. BURT P. J., YEN C., XU X., "Local Correlation Measures for Motion Analysis: A Comparative study.", Proceedings

IEEE Conference on Pattern Recognition and Image Processing , 269-274, 1982.

5. TAGUCHI A., TAKASHIMA H., MURATA Y., "Fuzzy Filters for Image Smoothing.", Proceedings SPIE Vol. 2180 Nonlinear Image Processing V. , 332-339, 1994.

6. BOZIC S. M., Digital and Kalman Filtering, 100-105, Edward Arnold, 1979.

7. BERTERO M., POGGIO T., TORRE V., "Ill Posed Problems in Early Vision." , Proceedings of the IEEE Vol 76 No. 8 , 869-889 August 1988.

8. ZADEH L., "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes" , IEEE Transactions on Systems Man and Cybernetics Vol. 3 No. 1 , 28-44 , January 1973.

Fuzzy Regularisation of Depth Maps Derived From Image Sequences

N. Rothwell Hughes, BSc, MSc, CEng, MIEE, MInstP.
University of Wales College Newport, UK
email: nrhughes@newport.ac.uk

Dr G. R. Wilson, BSc(Tech), PhD, CEng, MIEE.
University of Wales College Newport, UK
email: grwilson@newport.ac.uk

Dr G. N. Roberts, BSc, MSc, PhD, CEng, FIEE, FIMarE.
University of Wales College Newport, UK
email: g.roberts@newport.ac.uk

ABSTRACT

In this paper two types of nonlinear filter structure based on fuzzy logic are presented and applied to the problem of depth map regularisation. The first type of filter uses a fuzzy inferencing system to select the coefficients of a finite impulse response filter. The second uses a first order Sugeno fuzzy inferencing system to implement the filter. A correlation-based stereo matching algorithm is used to extract the depth information from a sequence of two dimensional images of a static scene under known camera motion. The resulting depth map is, however, corrupted by noise and areas of gross error due to the ill posed nature of the matching problem. The depth map is smoothed, thus regularising the problem, using the fuzzy-based filters which embody *a priori* assumptions about the underlying true depth map.

1. INTRODUCTION

The identification of the relative (and absolute) depth of points in a viewed scene is a potentially important low-level vision process in such applications as intelligent autonomous vehicle navigation, robot guidance, and industrial inspection in hazardous or inaccessible locations. This paper is concerned with this low-level vision process of depth reconstruction.

The basic principle of operation of the technique used is that of correlation-based stereo matching. The viewed scene is assumed to be static, and a sequence of images is taken by a single camera moving in a known way between images. The camera motion is such as to allow the use of the epipolar constraint, and the camera displacement is small enough between images in the sequence to restrict the search space for matching. This restricted search space provides a constraint which reduces the likelihood of a false match. However, small camera displacement (baseline) gives a depth measurement of low precision. In order to obtain the required precision the image displacements or disparities of corresponding points in the first and n^{th} image are tracked using a Kalman filter. The Kalman filter tracks the increasing disparity through the image sequence until the required precision in the depth measurement is attained. The resulting dense disparity map is converted into the corresponding depth map using the inverse perspective transformation. This depth map, which is a two dimensional (2-D) function aligned with the first image in the sequence, is corrupted by noise and areas of gross error.

These errors arise from the fact that, in common with other so called 'inverse problems', the problem of depth recovery is mathematically 'ill posed' [1]. In order to solve such ill posed problems it is necessary to impose constraints determining allowable solutions such that the best solution is chosen given uncertain and ambiguous measurements. A general approach to such problems is that of regularisation. In this approach the optimum solution is that which jointly minimises two energy or cost functionals. The first cost functional measures the closeness of a solution to the measurement, whilst the second measures how well the solution satisfies some *a priori* constraint such as smoothness. The second cost functional is often called the regularising operator or stabilising functional.

The choice of the stabilising functional is determined by the *a priori* beliefs about the underlying true solution and of confidence in those beliefs relative to the data. The minimisation of the functional is often carried out using iterative

gradient descent methods and can require much computing time. The process of minimising the energy functional can be thought of as a processing module which has as its inputs the measured disparities and the *a priori* beliefs about reasonable constraints on the smoothness of the disparity map; and has as its output the disparity map solution which best fits both the measured disparities and the smoothness constraints. Previous workers [2] have used smoothing stages based on the piecewise continuous generalised splines of Terzopolous [3] in order to regularise the problem of depth extraction from image sequences. A new approach based on fuzzy logic is proposed and explored in this paper. An advantage of the fuzzy approach is that the regularising *a priori* assumptions are explicitly embodied in the fuzzy rule base and the input and output membership functions.

The work described here compares two types of fuzzy-logic-based depth map regularisation stages. The first approach (see also [4]) uses a filter structured in a similar way to a 2-D finite impulse response filter, but in which the filter weights applied to each pixel in a 5x5 filter window are determined by the output of a fuzzy inferencing system. This general approach has been adapted from that used by Taguchi et al. [5] for the restoration of gray scale images.

The second approach is based on the first order Sugeno method of fuzzy inference [6]. This Sugeno-based fuzzy filter structure is suggested by the equivalence of the first order Sugeno output sets to finite impulse response (FIR) filters. By adopting the Sugeno structure the filter structure is completely contained within the fuzzy inferencing system (FIS). This results in a fuzzy filter stage which acts directly on the disparity data, instead of determining the weights of an adaptive filter as in the first type of filter. The structure of the second type of filter has the advantage that it allows the possibility of training the filter using simulated input-output data pairs.

Section 2 of the paper describes the algorithm used here to derive depth maps from image sequences. Section 3 describes the first type of fuzzy filter, and section 4 the second, Sugeno-based, filter. Section 5 presents some results using both types of filter on simulated depth maps and depth maps derived from real images. Finally section 6 presents the conclusions and future directions of the research.

2. ALGORITHM DESCRIPTION

The stages of the algorithm used to generate the disparity map are as follows:

(i) The first and second intensity images from the image sequence are read in as 256x256 matrices, each entry in the matrices corresponding to a gray scale value in the images

(ii) The Sum of Squared Differences (SSD) centred on each pixel (i,j) in image 1 and a 3x3 patch centred on pixels (i,j+shift) in image 2 are computed for a range of shifts. The SSD is a simple measure of match between the two patches.

(iii) The shift which generates the minimum SSD, 'best_shift', and its two neighbours 'best_shift ± 1' are stored along with their SSDs.

(iv) A quadratic is fitted to these three points. The minimum of the quadratic gives a sub pixel estimate of the shift or disparity for pixel (i,j). The width of the quadratic function (i.e. the coefficient of the square term) is taken as a measure of the uncertainty in the disparity measurement (*op cit*[2]).

(v) Steps (ii) to (iv) are then iterated with increasing matching patch size, until the uncertainty measure has dropped below a threshold and ceased to improve, or the matching patch exceeds a maximum size. The rate of increase of the patch size is controlled by the incremental improvement in the uncertainty.

(vi) The disparity with least uncertainty and the uncertainty measure are passed to a scalar Kalman filter. The filter tracks the evolution of the disparity for each pixel from image 1 to n which is modelled as a linear function of the camera shifts 1 to n. The prior disparity is assumed to be in the middle of the range of possible disparities, but with a high uncertainty, for all pixels

(vii) After the above steps are carried out for all pixels a disparity map, or inverse depth map, exists for all pixels. This disparity map is converted to the equivalent depth map using the inverse perspective projection and the known camera parameters. The depth map is then passed to a fuzzy logic filtering stage for smoothing, and the current disparity map is updated using the forward perspective projection. The final smoothed map is stored.

(viii) All the above steps are repeated with images 1 and n where n= 3,4,5... The search space along the scanline within which matches are sought for each pixel is centred on the predicted disparity based on the gradient of disparity with camera shift. At each stage the Kalman filter updates its estimate, and the fuzzy smoother smooths the depth map.

(ix) At the end of each match/smooth step a depth value has been obtained for each pixel in the original intensity image and a dense x,y,z map of the original scene can be recovered.

3. INDIRECT FUZZY FILTER

The first type of fuzzy filter, which is described in more detail in *op cit*[4], is structured in a similar way to a 2-D finite impulse response filter (figure1), but the filter weights applied to each pixel in a 5x5 filter window are determined by the output of a fuzzy logic system.

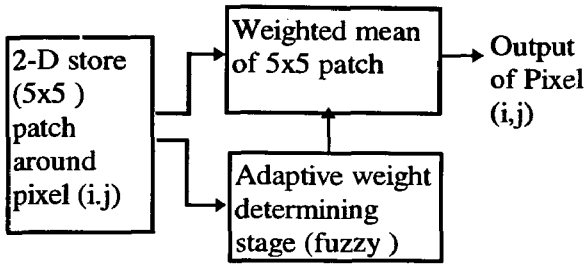

Figure 1: Adaptive Fuzzy Filter.

The crisp input values that are fuzzified are:

(i) the Euclidean distance of a pixel whose weight is being evaluated, Pixel($i+h,j+k$) from the pixel to which the smoothed disparity output will be assigned, Pixel(i,j). This distance is:

$$pixel\ dist = \sqrt{h^2 + k^2} \tag{1}$$

and
(ii) a measure of the difference in disparity of each pixel in the 5x5 filter window, $d(i+h,j+k)$ from the median of the disparities, *d_median*, in the window. This disparity difference measure is:

$$\left| \frac{d(i+h, j+k) - d_{median}}{divisor} \right| \tag{2}$$

where $divisor = d_{median}$ when $d_{median} > 1$, and $divisor = 1$ otherwise.

The fuzzy inferencing system to which the above observation variables are applied is a zero-order Sugeno system. Both fuzzy observation variables have three triangular membership function fuzzy sets. There are nine rules in the rulebase, and the 'minimum' operator is used for the 'AND' conjunction as well as for the implication operator, 'THEN', in the fuzzy rulebase. The firing weights for each rule are applied to the appropriate one of five Sugeno singleton output sets, O_i. These are used rather than the more common Mamdani fuzzy output sets for computational simplicity. The aggregate output $O(h,k)$ for each pair of inputs is obtained as a normalised weighted sum over all the rules, with the weights, w_i being the firing weight for each rule.

$$O(h,k) = \sum_{i=1}^{R} \frac{w_i O_i}{w_i} \tag{3}$$

This crisp output is now taken as the weight of pixel (i+h,j+k). The process of obtaining filter weights is repeated for all the pixels in the 5x5 filter window. The final filtered output, $d_{fil}(i,j)$ which replaces the unfiltered value of disparity for pixel (i,j) is calculated from:

$$d_{fil}(i,j) = \frac{\sum_{h=-2}^{h=+2} \sum_{k=-2}^{k=+2} O(h,k) d(i+h, j+k))}{\sum_{h=-2}^{h=+2} \sum_{k=-2}^{k=+2} O(h,k)} \tag{4}$$

The first type of fuzzy smoother acts indirectly on the data and adds an additional stage between the input corrupted disparity data and the output smoothed map. This makes it difficult to use existing techniques of input-output data pair training of the fuzzy system. However, after adjustment of the parameters of the fuzzy inference system using simulated test data, this type of filter works effectively on depth maps derived from real image sequences. The filter smooths the depth map whilst preserving discontinuities. The effect of the fuzzy weight-determining stage is to assign a larger weight to pixels closer to the pixel being filtered, and to those whose disparity is closer to the median of the filter window.

4. SUGENO-BASED FILTER

The structure of the second direct-acting fuzzy filter is based on a first order Sugeno type Fuzzy Inferencing System (FIS) *op cit* [6]. This type of FIS differs from the Mamdani type of FIS in that the values of the output sets are a linear combination of the crisp inputs, x_i. Each output set is defined by its coefficient set, a_i.

$$O_i = a_0 + \sum_{i=1}^{i=R} a_i x_i \tag{5}$$

If the coefficients a_i are chosen appropriately, each output set can be regarded as a transversal or finite impulse response (FIR) filter. Thus the Sugeno FIS can be drawn as a feedforward nonlinear filter structure (figure 2).

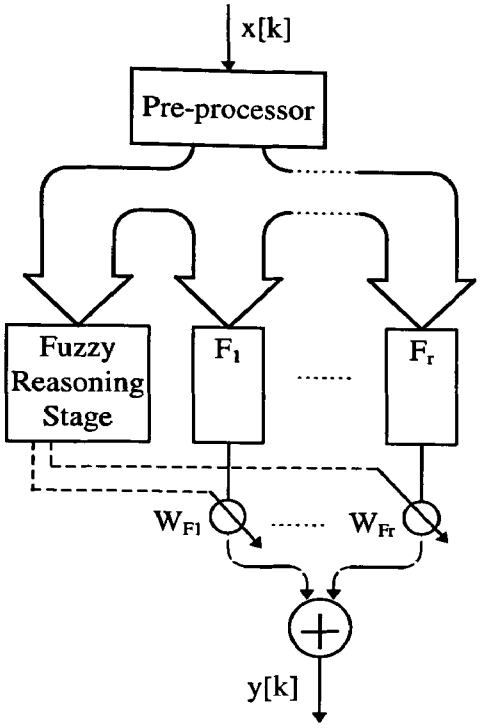

Figure 2. Sugeno-based filter

The pre-processor assembles the raw depth data samples from the filter window into a correctly ordered vector for input to the Sugeno FIS. The blocks F_1 to F_r represent the output sets of the first order Sugeno network. There are as many of these blocks as there are output sets in the Sugeno network. The weights W_{F1} to W_{Fr} which are applied to the outputs of the linear filters are the firing weights of each rule for the applied inputs x[k]. The final output of the filter is the weighted sum of the outputs of the filter bank with the weights determined by the fuzzy stage as a non-linear function of the pre-processed input. Viewed in this way it can be seen that the first order Sugeno network can implement a general non-linear filter. This structure can implement any linear FIR filter as a special case by fixing the output of the fuzzy inferencing stage to select the appropriate filter or combination of filters regardless of the input. However, the advantage of this filter structure lies in its ability to choose the output coefficient set, and hence the filter action, depending on the inputs to the filter. This leads to an overall non-linear filter which in effect interpolates multiple linear filters. The choice of the coefficients for the different output sets can thus be guided by conventional filter design techniques. It is also easier to train the filter output coefficients and the input sets' membership functions using simulated input-output data with this second fuzzy filter structure than with the first type of fuzzy filter.

Let there be N inputs defined for the fuzzy system, and let there be R rules. Also let the pre-processor assemble the input data into the 1xN input vector $\boldsymbol{X}$, and let the feedforward fuzzy stage be represented by the function $f(.)$ which maps the 1 x N vector $\boldsymbol{X}$ into the 1xR rule firing-weight vector $\boldsymbol{W}$. Assume without loss of generality that each rule has a corresponding filter in the filter bank. If the coefficients of the R filters in the filter bank are represented by the RxN matrix $\boldsymbol{A}$, the action of the filter can be described by the equation:

$$y[k] = \boldsymbol{W.A.X}^T = f(\boldsymbol{X}).\boldsymbol{A.X}^T \qquad (6)$$

The function $f(.)$ is dependent on the choice of fuzzy rules and input fuzzy sets.

If the r^{th} fuzzy rule in a rulebase of R rules is :

$$IF\ x_1\ is\ FS_1^r\ AND \ldots . x_N\ is\ FS_N^r\ THEN\ W_r\ is\ w^r \qquad (7)$$

where the x_n are elements of the crisp input vector $\boldsymbol{X}=(x_1 \ldots x_N)$, the W_r are elements of the weight vector $\boldsymbol{W}=(W_1 \ldots W_r \ldots W_R)$, and the FS are fuzzy sets defined by Gaussian membership functions, then the r^{th} element in the weight vector is given by:

$$W_r = \frac{min\{\mu_{FS1}^r(x_1),\ldots.\mu_{FSN}^r(x_N)\}}{\sum_{r=1}^{r=R} min\{\mu_{FS1}^r(x_1),\ldots.\mu_{FSN}^r(x_N)\}} \qquad (8)$$

where $\mu_{FSn}^r(x_n)$ is the membership of x_n in the fuzzy set FS_n^r.

The nonlinear feedforward mapping, $f(.):\boldsymbol{X}\rightarrow\boldsymbol{W}$ is determined by equations (7) and (8). In order for the filter defined by equations (6)-(8) to be useful in depth map regularisation appropriate rules and input fuzzy sets have to be identified. However, for a 3x3 two dimensional filter with two fuzzy sets per input, an exhaustive set of rules requires 2^9 rules. Therefore an effective filter has to be generated using only a small subset of all the possible rules. The results for Sugeno-based filters, described in the next section, are derived from rules which aim to identify depth discontinuities or edges within a 3x3 and 5x5 filter window. A region within the window to which the centre pixel belongs is inferred from these edges. The centre pixel

value in the filter window is then replaced by the arithmetic mean of this region. The formation of the arithmetic means for each rule is carried out by the filter blocks in figure 2.

5 RESULTS

5.1 Indirect fuzzy filter

The indirect fuzzy filter parameters were adjusted on depth map test data derived from a sequence of random dot images. A square area in these images was shifted by a fixed known amount from one image to the next in the sequence. The mean squared error (MSE) of the resulting depth map after filtering was used as the performance criterion. After adjustment of the fuzzy inferencing system parameters, the control map for the weights $O(h,k)$ in equation (3) was as shown in figure 3. It can be seen from figure 3 that small departures from the normalised median are heavily penalised, and that pixels further from the centre pixel are assigned a lower weight.

Figure 4 shows the first image in a sequence of real off-camera images. Figure 5 is the resulting depth map after ten four mm shifts of the camera if no regularisation is carried out. Figure 6 is the depth map which results from the same image sequence when smoothed between matches by the indirect fuzzy filter.

5.2 Sugeno-based fuzzy filter

The Sugeno-based fuzzy filter was tested for its filtering action on a simulated 'depth' map consisting of a pyramid, a stepped pyramid, a Gaussian 'hump' , and several square columns of different widths and heights ranging from 1.0 to 2.0. This depth map was corrupted with zero mean, variance 0.16 Gaussian noise. 3x3 and 5x5 Sugeno-based filters of nineteen rules were applied to this depth map and the results compared with 3x3 and 5x5 moving average and median filters. A comparison of their performances on the simulated depth map is shown in table 1.

The performance of the fuzzy filters on this test data gives an indication of the potential of such filters. Figure 7 shows a depth map smoothed by a 5x5 Sugeno-based filter. This depth map is not as well smoothed as the one generated by the indirect fuzzy filter, however the performance of the Sugeno-based filter has the potential to be improved using input-output training data pairs.

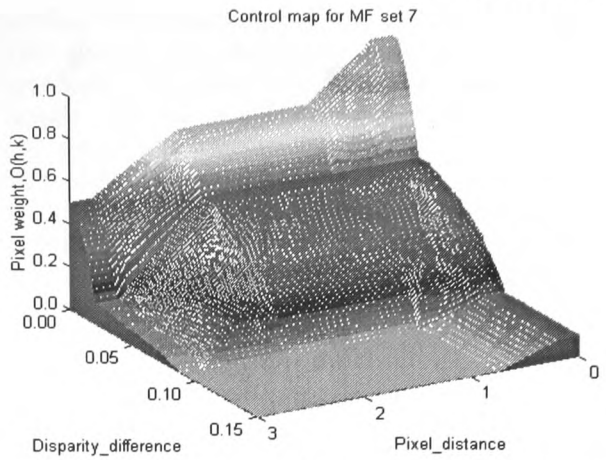

Figure 3. Control map for indirect fuzzy filter

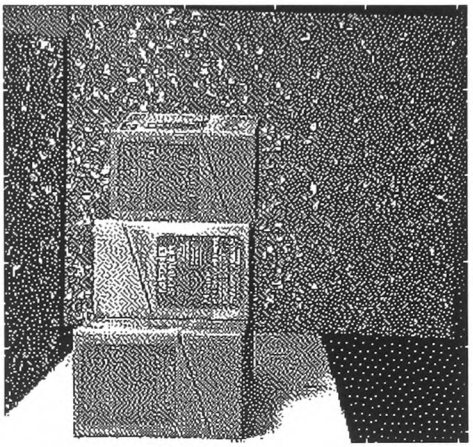

Figure 4. First image from sequence'Boxes

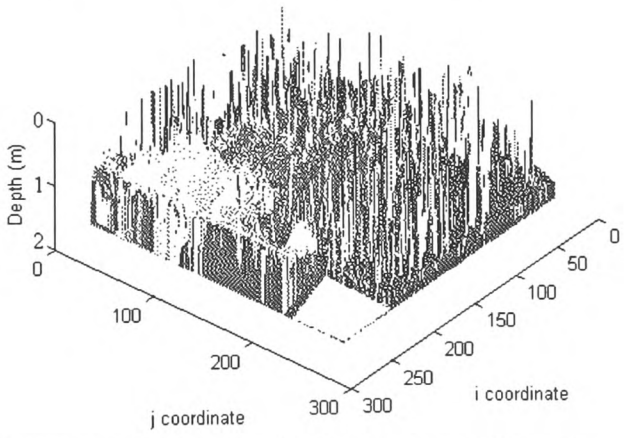

Figure 5. Depth map from sequence 'Boxes' generated by algorithm with no regularisation

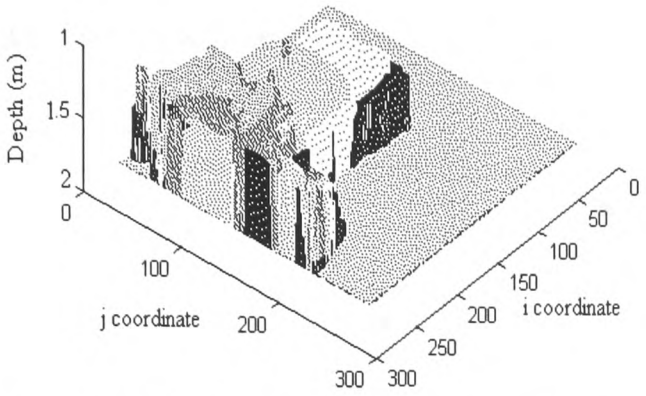

Figure 6. Depth map of test image 'Boxes' smoothed by indirect fuzzy filter

Table 1. Comparison of filter performance on test data corrupted with Gaussian noise

Filter type	Mean squared error
3x3 median	0.0287
3x3 moving average	0.0214
3x3 Sugeno-based	0.0207
5x5 median	0.0145
5x5 moving average	0.0131
5x5 Sugeno-based	0.0105

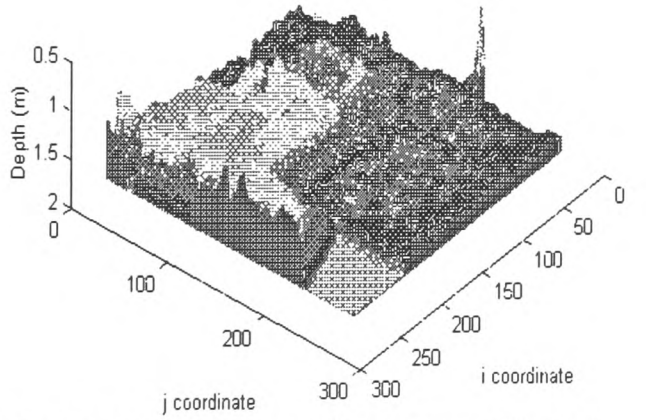

Figure 7. Depth map of test image 'Boxes' smoothed by Sugeno-based fuzzy filter

8. CONCLUSIONS AND FUTURE WORK

Two different fuzzy-logic-based methods of regularising the noisy depth maps resulting from a correlation-based stereo matching algorithm have been presented. The first of these methods, the indirect method, results in good smoothing of the depth map, whilst preserving edges. The second method, based on the first-order Sugeno fuzzy inferencing method, encapsulates the whole filter within the fuzzy inferencing system, and therefore allows the use of input output training data pairs. The number of possible rules for a two dimensional filter grows exponentially with the filter size, therefore a subset of the possible rules has to be chosen. Initial results on simulated depth maps corrupted with Gaussian noise using such a filter show a small improvement over moving average and median filters. Although the performance on real depth maps is not yet as good as the indirect acting fuzzy filter, the performance can be improved using representative training data.

Work is currently in progress to improve the performance of the Sugeno-based filter. This will be achieved by using training data corrupted with both impulsive and Gaussian noise to identify effective rules; and also by using simulated annealing to train the input fuzzy set parameters on the same type of training data.

REFERENCES

[1] Bertero, M. Poggio, T. and Torre V. *ill Posed Problems in Early Vision*, Proceedings of the IEEE Vol 76 No. 8 pp 869 - 889, Aug. 1988.

[2] Matthies, L., Kanade, T. and Szeliski R. *Kalman Filter-based Algorithms for Estimating Depth from Image Sequences,* International Journal of Computer Vision. No. 3 pp 209-236, 1989

[3] Terzopoulos D. *Regularisation of Inverse Visual Problems Involving Discontinuities,* IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 8 No. 4 pp 413 - 424, July 1986

[4] Rothwell Hughes, N., Wilson, G. R., and Roberts, G. N. *Kalman and Fuzzy Logic Filters for 3-D Industrial Inspection Using a Single Camera.* Proceedings of the International Conference on Recent Advances in Mechatronics (ICRAM95) Vol. 2 pp 595-600, August 1995.

[5] Taguchi, A., Takashima, H. and Murata, Y. *Fuzzy Filters for Image Smoothing*, SPIE Vol. 2180 Non-linear Image Processing V pp 332 - 339, 1994.

[6] Tagaki, T. and Sugeno, M. *Derivation of Fuzzy Cntrol Rules from Human Operator's Control Actions.* Proceedings of the IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis, pp 55-60, July 1983.

APPLICATION OF FUZZY SIGNAL PROCESSING TO THREE DIMENSIONAL VISION

N Rothwell Hughes, G N Roberts ,G R Wilson.

Mechatronics Research Centre
University of Wales College, Newport. UK

Introduction

The ability to sense and construct a three dimensional representation of the operating environment is an important enabling technology for intelligent autonomous vehicle navigation, robot guidance, and industrial inspection in hazardous or inaccessible locations. The goal of recovering three dimensional shape information from multiple two dimensional images also offers the achievement of completely automatic inspection of three dimensional manufactured objects. The use of CCD cameras and correlation-based stereo matching techniques is attractive because dense depth maps of the viewed scene can be generated in this way.

However, the recovery of depth using correlation-based matching is a mathematically ill-posed problem, Bertero et al (1), for which *a priori* information about the form of acceptable depth map solutions is required to correct erroneously matched areas of the depth map. This is usually referred to as 'regularisation' and is closely related to the problem of filtering corrupted signals. An important feature of any regularisation stage for depth maps is the ability to identify and deal appropriately with discontinuities. The use of fuzzy-based smoothing filters offers a new approach to the regularisation of depth maps.

Work on the use of fuzzy logic in filtering of one dimensional signals in the presence of noise has been reported by Kim and Kosko (2), Kwan and Kai (3), and Hsaiao and Lai (4). Other workers, Taguchi et al (5), Takashima et al. (6), Yang and Toh (7), Peng and Lucke (8), and Arakawa (9) report work on fuzzy logic-based filters for image restoration. Similar techniques for spatio-temporal filtering of impulsive noise corrupted TV signals have been reported by Mancuso et al (10). The authors have applied similar techniques to the smoothing of depth maps derived from correlation-based matching algorithms, Rothwell Hughes et al. (11,12). The motivation for much of the work listed above is to develop a non-linear filter system which can cope with signals corrupted with both Gaussian and impulsive noise whilst preserving signal discontinuities. An advantage of the fuzzy approach is that the *a priori* assumptions are explicitly embodied in the fuzzy rule base and the input and output membership functions. In the case of filtering depth maps which have been derived from sequences of grey level images, information over and above the raw depth data is available which can be input to the filter. This additional information is the uncertainty in the depth measurement at each pixel and the original grey scale images

The remainder of the paper consists of six further sections. The first describes the basic depth from image sequences algorithm The next section describes the basic architecture of the fuzzy filters used to smooth the depth maps. The three following sections describe the development of three types of fuzzy filter of increasing complexity. Each filter builds on the previous filter by incorporating more information as input to the fuzzy inferencing system. The results of applying the three types of fuzzy filter to depth maps derived from simulated images and to sequences of real images is presented and discussed in these sections. The final section is a conclusion and discussion of the results obtained.

Depth From Image Sequences

The basic principle of operation of the technique used is that of correlation-based stereo matching. The viewed scene is assumed to be static, and a sequence of images is taken by a single camera moving in a known way between images. The camera motion is such as to allow the use of the epipolar constraint, and the camera displacement is small enough between images in the sequence to restrict the search space for matching. In order to obtain the required precision the image displacements or disparities of corresponding points in the first and n^{th} image are tracked using a Kalman filter. The Kalman filter tracks the increasing disparity through the image sequence. This technique is similar to that described in Matthies et al (13), and Trucco et al (14). Details of the Kalman filter used are given in (11).

The stages of the algorithm used to generate the disparity map are as follows:

(i) The first and second intensity images from the image sequence are read in as matrices of pixel grey level, each entry in the matrices corresponding to a grey scale value in the images

(ii) The Sum of Squared Differences (SSD) centred on each pixel (i,j) in image 1 and a 3x3 patch centred on pixels (i,j+shift) in image 2 are computed for a range of

shifts. The SSD is a simple measure of match between the two patches, Annandan (15).

(iii) The shift which generates the minimum SSD, 'best_shift', and its two neighbours 'best_shift ± 1' are stored along with their SSDs.

(iv) A quadratic is fitted to these three points. The minimum of the quadratic gives a sub pixel estimate of the shift or disparity for pixel (i,j). The width of the quadratic function (i.e. the coefficient of the square term) is taken as a measure of the uncertainty in the disparity measurement, (14).

(v) The disparity with least uncertainty and the uncertainty measure are passed to a scalar Kalman filter. The filter tracks the evolution of the disparity for each pixel from image 1 to n which is modelled as a linear function of the camera shifts 1 to n. The prior disparity is assumed to be in the middle of the range of possible disparities, but with a high uncertainty, for all pixels

(vi) After the above steps are carried out for all pixels a disparity map, or inverse depth map, exists for all pixels. This disparity map is converted to the equivalent depth map using the inverse perspective projection and the known camera parameters. This depth map, which is a two dimensional (2-D) function aligned with the first image in the sequence, is corrupted by noise. The depth map is then passed to the filtering stage for smoothing, and the current disparity map is updated using the forward perspective projection. The final smoothed map is stored.

(vii) All the above steps are repeated with images 1 and n where n= 3,4,5... The search space along the scanline within which matches are sought for each pixel is centred on the predicted disparity based on the gradient of disparity with camera shift. At each stage the Kalman filter updates its estimate, and the fuzzy smoother smooths the depth map.

(viii) At the end of each match/smooth step a depth value has been obtained for each pixel in the original intensity image and a dense x,y,z map of the original scene can be recovered.

Filter Architecture

All the filters described in this paper share a common architecture. This architecture is based on the Sugeno Fuzzy Inferencing technique described by Tagaki and Sugeno (16). In this type of Fuzzy Inferencing System (FIS) each element of the crisp input data vector is assigned a degree of membership in a set of fuzzy input sets. This converts the crisp input data into a fuzzy data set a process often called 'fuzzification'. This fuzzy data set is fed to an inferencing engine. The inferencing engine produces a firing weight for each one of a set of rules in a rulebase. These rules link patterns in the fuzzy input sets to an output set. In the Sugeno type of FIS the r^{th} output set is a function F_r of the crisp input data. In the case of the filters described here the output functions are linear combinations of the input data. If the n input data vector elements are described by x_i , the j^{th} output set O_j is defined by its coefficient set, a_{ji}. There are R rules and R output sets so that for each rule:

$$O_j = a_0 + \sum_{i=1}^{i=n} a_{ji} x_i \qquad (1)$$

Each rule is fired to a degree w_j depending in a non-linear way on the data. The form of the non-linearity is dependent on the fuzzy inferencing engine and its parameters. The final crisp output y of the FIS is the weighted sum over all R rules:

$$y = \frac{\sum_{j=1}^{j=R} w_j O_j}{\sum_{j=1}^{j=R} w_j} \qquad (2)$$

The overall Sugeno FIS can be drawn as a mapping from an input vector X to a single crisp output Y as shown below in figure 1. Such a mapping can be regarded as a filter. In the case of the filters described here the pre-processor stage collects the depth data from a 3x3 filter window into a vector. This vector is concatenated with other available data. This other data can incorporate additional information such as the associated uncertainty in the depth data. It can also be extracted from the depth data such as the median of the depth data in the filter window. The weights W_j shown in Figure 1 are the normalised firing weights:

$$W_j = \frac{w_j}{\sum_{j=1}^{j=R} w_j} \qquad (3)$$

The filters described in the following section all use the architecture described above. They differ from one another in the data which is input to the filter. As additional data is added to the input vector, the information from this additional data is used to modify the action of the fuzzy filter. In effect the fuzzy filter is integrating available information.

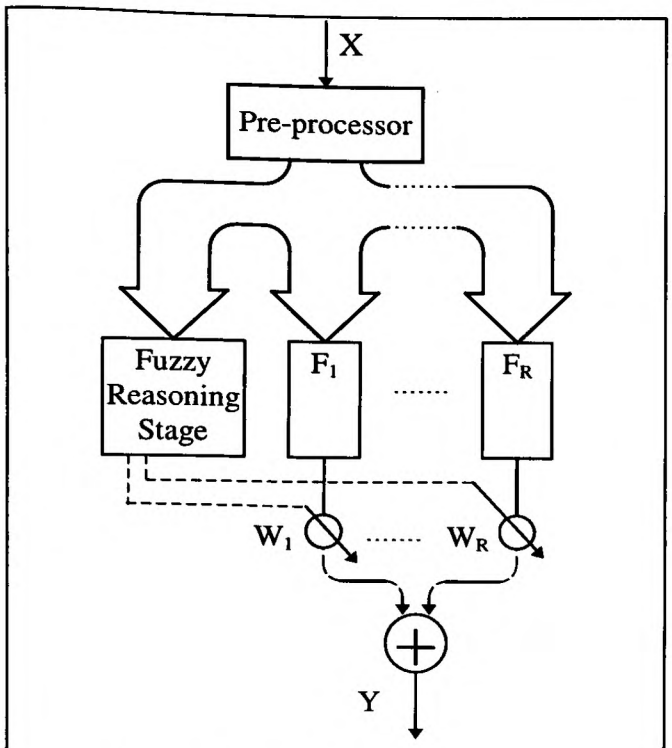

Figure 1: Fuzzy filter architecture

Fuzzy Difference from Median filter (DMF)

The first type of filter uses the fact that median filters are effective in smoothing impulsive noise whilst preserving discontinuities. They are less effective in a mean squared error sense when used on areas of images which are smooth (do not contain discontinuities) and corrupted with more Gaussian noise. Simulated image sequences of moving patches of different grey level and texture have been generated to investigate the behaviour of the SSD matcher. The ground truth depth maps of these simulated image sequences are therefore known and do not depend on camera and viewed scene calibration. Examination of histograms of errors in depth maps generated from simulated image sequences suggest that the noise process is a mixture of Gaussian and more impulsive noise and that the noise process differs in different parts of the image.

The first filter strategy adopted is therefore to modify the filter behaviour between a median filter and a moving average filter in order to achieve an improved performance over either a pure median or moving average filter. The performance measure adopted is that of root mean squared error (RMSE) over the depth map for a depth map derived from a simulated image sequence. The use of RMSE as a performance measure does not perhaps attach sufficient weight to preservation of depth discontinuities, but allows some objective comparison of the effect of the filter to be made.

The first image of the test image sequence is shown below in figure 2. The ground truth depth map for this image sequence is shown in figure 3.

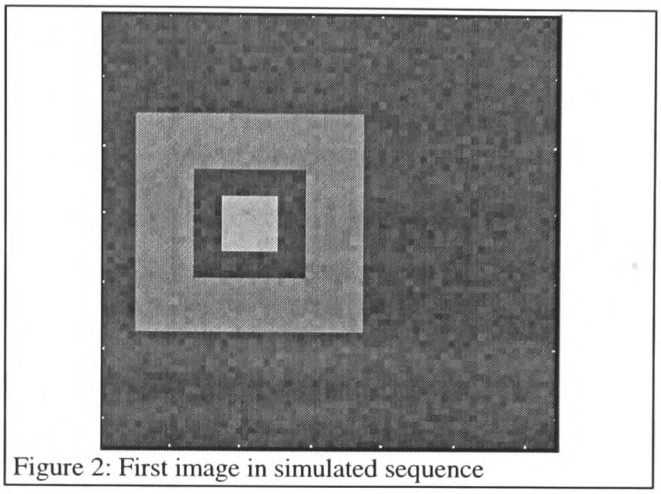
Figure 2: First image in simulated sequence

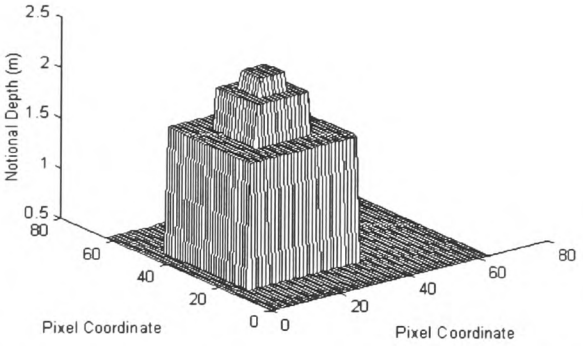

Figure 3. True depth map for simulated image sequence

The input vector to the fuzzy system consists of the nine pixels reading from left to right and down in a 3x3 filter window; the difference from the filter window median for each of the nine pixels, and the median itself. Thus the crisp input vector has 19 elements. The difference from median inputs are classified by their membership in three fuzzy sets labelled negative large (NL), positive large (PL) and small (S). The membership functions of these sets are Gaussian in shape and are parameterised by two parameters, width and centre.

The rules for each pixel (l,m) in the filter window having depth d(l,m) are of the form:

IF Difference-From-Median is S THEN output is : d(l,m)

IF Difference-From-Median is NL or PL THEN output is : Median.

The output sets are achieved by setting the output set coefficient a_{ji} corresponding to the input values to be one. The effect of this difference-from-median filter (DMF) is to act as a moving average filter in smooth areas and as a median filter in areas containing discontinuities.

After adjustment of the input set parameters the root mean square errors achieved after an input sequence of ten simulated images were as shown in table 1 below.

Table 1. Comparison of RMSE for filters

Filter Type	RMSE on simulated image
Moving Average (3x3)	0.2482
Median (3x3)	0.2589
DMF (3x3)	0.2303

The Resulting depth map is shown in figure 4 below. It can be seen that there is some loss of definition at the depth discontinuities, although the RMSE is better than either the moving average or the median filters. The discontinuities can be better preserved by adjusting the input set parameters, but only at the cost of worsening RMSE.

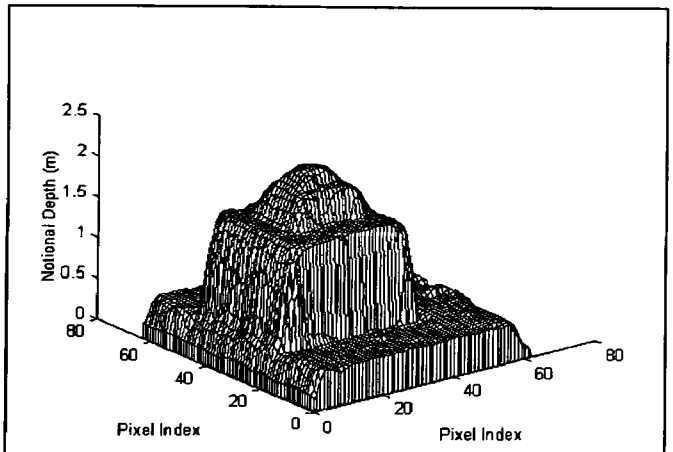

Figure 4. Depth map filtered by Fuzzy Difference-From-Median Filter

Fuzzy DMF with uncertainty measure input

The second type of filter utilises the fact that the SSD matcher that is used can produce an estimate of uncertainty in its measurements of depth (13). This uncertainty measure or quasi-variance is used by the Kalman filter to integrate the depth measurements through the sequence of images. It can also be used to weight the importance of each pixel in a spatial filter window. The uncertainty measure for each pixel in the filter window (a nine element vector of uncertainty measures) is concatenated with the vector of input values used by the fuzzy filter. This results in an input vector whose first eighteen elements are the same as the first eighteen for the fuzzy DMF, the next nine inputs are the uncertainty measures for each pixel and the last element is the median of the depths in the filter window.

The work described in (14) reveals some flaws in the uncertainty measure used in this algorithm and a better empirically derived uncertainty measure is proposed there. Clearly the uncertainty measure needs to be reliable in order for its use to be effective. Because the ground-truth depth map is known for the simulated image, the actual error at each pixel can be compared to the estimated uncertainty. Figure 5 below shows the cross correlation of the measured uncertainty and true error for a portion of a depth map, which shows that the uncertainty measure is quite well correlated with the actual error.

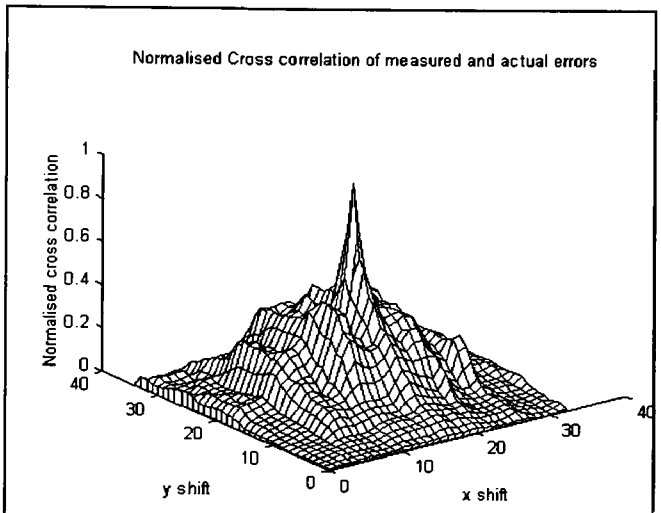

Figure 5. Cross correlation of actual error and estimated uncertainty measure at each pixel.

Only one fuzzy set is defined for the uncertainty level input. This has a Gaussian membership function labelled Small (S). The rule base is now modified so that the rules are of the form:

IF Difference-From-Median is S AND the uncertainty is S THEN output is : d(l,m)

IF Difference-From-Median is NL or PL THEN output is : Median.

This change to the first rule has the effect of strengthening the contribution of pixel depth values whose measured uncertainty is low.

With this change to the rules and after adjustment of the fuzzy set parameters a Root Mean Square Error for the depth map derived from simulated images of 0.2286 was achieved. This is a small improvement on the RMSE achieved by the fuzzy DMF. The associated depth map using this smoothing stage is shown in Figure 6.

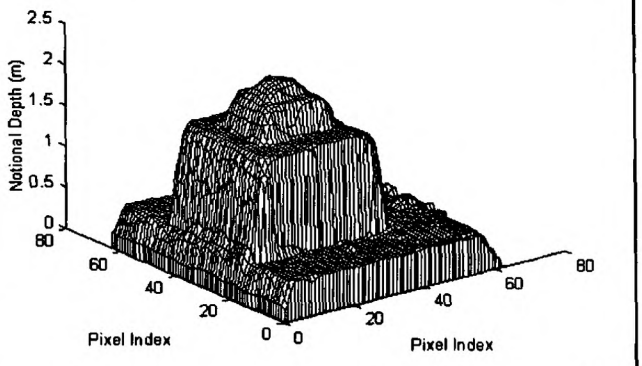

Figure 6. Depth map after smoothing with Fuzzy DMF with uncertainty input.

Figure 7 shows the first image of a sequence of real images of three stacked computer-disk boxes. The tonal range in the image has been severely reduced for reproduction. The lower box is about a metre from the camera and each box is 0.1m further from the camera than the one below. Figure 8 shows the depth map derived from this sequence of real images after smoothing with the fuzzy DMF with uncertainty input rendered as a grey scale image.

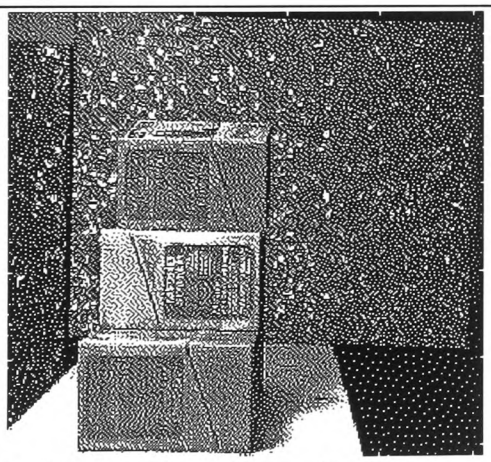

Figure 7. First image in sequence of real images

Areas in the grey-scale-rendered depth map which are closer to the camera are represented by a lighter shade of grey. The dark patch on the lowest of the disk boxes in the depth map represents a failure by the SSD matcher to match corresponding points between the image pairs. This is due to the lack of texture in these areas on the original images. The fuzzy filter is unable to fill in these large areas of failure. However the depth map is quite well smoothed by the filter in other areas and the depth discontinuities are preserved.

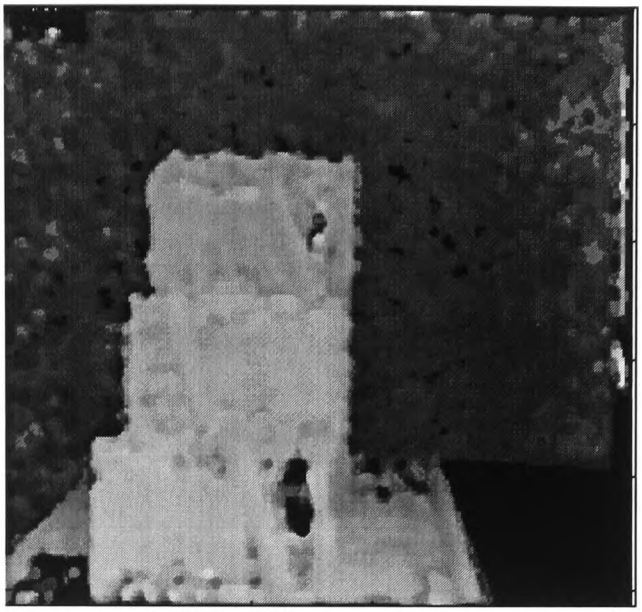

Figure 8. Depth map for real image sequence after smoothing with Fuzzy DMF with uncertainty input.

Fuzzy DMF with uncertainty input and edge input

The third type of filter attempts to use the edges extracted from the first grey scale image in the sequence as a guide to the location of edges in the depth map. Edges in grey scale images arise along lines of depth discontinuity. However, they also arise due to shadows, variations in reflectance, and variations in texture. Thus a grey scale edge offers supporting evidence of the existence of a depth edge at a location where there is some evidence of a depth discontinuity. However, the existence of a grey level edge does not of itself suggest a depth discontinuity at a location.

In order to incorporate this idea into the fuzzy DMF filter, a Sobel edge detector is applied to the first grey scale image in the sequence to form an edge map. The output of the Sobel operator is not thresholded, but left as a measure of edge strength. Since the depth map is aligned with the first grey scale image, it is also aligned with the edge map. Thus pixels in the depth map have corresponding pixels in the edge map. The corresponding edge map pixels for each 3x3 window are added to the vector which is input to the fuzzy filter. There is only one fuzzy set defined for the edge strength, which has a Gaussian member function with a maximum at 63 and a width parameter of 20.

The additional rules which are added to the rule base are of the form:

If the difference from median is NL or PL and the edge strength is large then output the median.

This rule assumes that the difference from median will tend to be large at depth discontinuities. If a grey scale edge is present where the difference from median is large then its presence will reinforce the filter's tendency to act as a median filter. Where there is little or no edge strength this rule will not be fired and will thus not reinforce the filter's tendency to act as a median filter.

The result of using this third type of filter to the simulated image sequence is shown below in Figure 9. The RMS error is poor at 0.449, although the depth discontinuities are distinct.

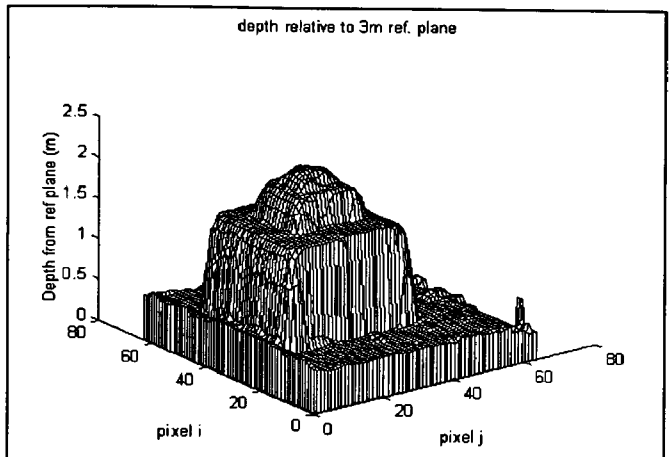

Figure 9. Depth map generated by DMF with uncertainty and edge input.

Conclusions

Three types of filter based on Sugeno fuzzy systems have been presented. These filters aim to improve on the results of a depth from image sequences algorithm using either a moving average or median filter as a depth map smoother. The fuzzy filters attempt to do this by using additional information on the uncertainties in the depth map and information on edge location in the original grey scale images. The results of the first two types of filter described show an improvement in RMS error on sequences of simulated images. The results using the edge information are disappointing, but work is ongoing to investigate the reasons for the poorer RMS error when this type of filter is used.

References

1. Bertero M, Poggio T and Torre V, 1988 , "Ill Posed Problems in Early Vision", Proc IEEE , 76 , 869 - 889.

2. Kim H M and Kosko B, 1996, "Fuzzy Prediction and filtering in impulsive noise", Fuzzy Sets and Sys , 77 , 15 -33.

3. Kwan H K and Cai Y, 1993, "Median Filtering Using Fuzzy Concept" , Proc IEEE Symp Circs and Sys , 2 , 824 - 827.

4. Hsaiao C Y and Lai C C, 1995, "Analysis and Design of Fuzzy Filter Algorithms", Proc IEEE/IAS Conf Ind Automation and Control ,1 , 413-420

5. Taguchi A, Takashima H and Murata Y, 1994, "Fuzzy Filters for Image Smoothing" , SPIE Conf Nonlinear Image Processing V , 1 ,332-339.

6. Takashima H, Taguchi A and Murata Y, 1995, "Edge Preserving Smoothing using the Fuzzy Control Technique " , Elec Comm J Pt III 78(1) , 43-72.

7. Yang X and Toh P S, 1995 "Adaptive Fuzzy Multilevel Median Filter", IEEE Trans Image Processing 4 , 680-682.

8. Peng S and Lucke L, 1995, "Multilevel Adaptive Fuzzy Filter for Mixed Noise Removal ", IEEE Int Symp Circs Sys, 1524-1527.

9. Arakawa K, 1996, "Median Filter based on Fuzzy Rules and its Application to Image Restoration", Fuzzy Sets and Sys , 77 , 3-13.

10. Mancuso M, De Luca R, Poluzzi R and Rizzotto G, 1996, "A Fuzzy Decision Directed Filter for Impulsive Noise Reduction", Fuzzy Sets and Sys , 77 , 111-116.

11. Hughes N, Wilson G and Roberts G, 1995, "Kalman and Fuzzy Logic Filters for 3-D Industrial Inspection Using a Single Camera", Int Conf Recent Advances in Mechatronics 2 ,595-600.

12. Hughes N, Wilson G and Roberts G, 1995, "Fuzzy Regularisation of Depth Maps Derived from Image Sequences", Int Conf On Mechatronics and Machine Vision in Practice 2 , 55-60.

13. Matthies L, Kanade T and Szeliski R, 1989, "Kalman Filter Based Algorithms For Estimating Depth From Image Sequences", Int Journ Comp Vision, 3 , 413-424

14. Trucco E, Roberto V, Tinonin S and Corbatto M, 1996, "SSD Disparity Estimation for Dynamic Stereo", Proc British Machine Vision Conference 1, 343-352.

15. Annandan P, 1989, "A Computational Framework and an Algorithm for Measurement of Visual Motion" , Int. Journ Comp Vision 2, 283-310.

16. Tagaki T and Sugeno M, 1983, "Derivation of Fuzzy Control Rules from Human Operator's Control Actions " Proc IFAC Symp Fuzzy Information Knowledge Representation and Decision Analysis , 55-60.